5-2017

Network Coding for Packet Radio Networks

Siddhartha S. Borkotoky
Clemson University, sborkot@g.clemson.edu

Network Coding for Packet Radio Networks

A Dissertation
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
Electrical Engineering

by
Siddhartha S. Borkotoky
May 2017

Accepted by:
Dr. Michael B. Pursley, Committee Chair
Dr. Daniel L. Noneaker
Dr. Harlan B. Russell
Dr. James J. Martin

Abstract

We present methods for network-coded broadcast and multicast distribution of files in ad hoc networks of half-duplex packet radios. Two forms of network coding are investigated: fountain coding and random linear network coding. Our techniques exploit the broadcast nature of the wireless medium by permitting nodes to receive packets from senders other than their designated relays. File transfer is expedited by having multiple relays cooperate to forward the file to a destination. When relay nodes apply fountain coding to the file, they employ a simple mechanism to completely eliminate the possibility of sending duplicate packets to the recipients. It is not necessary for the nodes to transmit multiple packets simultaneously or to receive packets from multiple senders simultaneously. To combat the effects of time varying propagation loss on the links, each sender has the option to adapt the modulation format and channel-coding rate packet-by-packet by means of an adaptive transmission protocol. We use simulations to compare our network-coded file distributions with conventional broadcast and multicast techniques that use automatic repeat request (ARQ). Our numerical results show that the proposed strategies outperform ARQ-based file transfers by large margins for most network configurations. We also provide analytical upper bounds on the throughput of file distributions in networks comprising four nodes. We illustrate that our network-coded file-distribution strategies, when applied to the four-node networks, perform very close to the bounds.

Acknowledgments

I would like to thank my advisor, Dr. Michael B. Pursley, for his guidance and support throughout my graduate studies at Clemson University. I am grateful to Dr. Daniel L. Noneaker, Dr. Harlan B. Russell, and Dr. James J. Martin for serving on my committee and providing their valuable feedback. I would also like to thank the Army Research Office for providing financial assistance to support my research.

I have immensely benefitted from my interactions with many of the current and former students at Clemson University. Special thanks to my collaborators Jason Ellis, Michael Juang, Sneha Kottapalli, and Michael Dowling for many helpful discussions on various research topics.

Finally, I would like to thank my parents and my sister for their love and constant support.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

In an ad hoc packet radio network, wireless nodes cooperate to distribute information without relying on fixed infrastructure or centralized control. The information to be disseminated is divided into packets and routes are established between source-destination pairs. Nodes along a route act as relays, receiving packets from the source and forwarding them towards the destination.

Automatic repeat request (ARQ) has been traditionally employed for reliable delivery of information in networks [1]. In a system that employs ARQ, the recipient must provide feedback to the sender acknowledging the receipt of the packets. The sender retransmits every unacknowledged packet until the packet is correctly received. While ARQ guarantees reliability, it has several drawbacks, especially when packets must be delivered to multiple recipients. Two major shortcomings are reduction in the throughput due to excessive retransmissions and the problem of feedback implosion. The authors of [1] state:

> "However, for one-to-very-many reliability protocols, ARQ has limitations, including the feedback implosion problem because many receivers are transmitting back to the sender, and the need for a back channel to send these requests from the receiver. Another limitation is that receivers may experience different

> loss patterns of packets, and thus receivers may be delayed by retransmission of packets that other receivers have lost that but they have already received. This may also cause wasteful use of bandwidth used to retransmit packets that have already been received by many of the receivers."

An alternative to ARQ is a class of packet-level erasure-correction coding schemes known as *linear network coding*, which includes *fountain coding* (FC) [2] and *random linear network coding* (RLNC) [3]. Instead of sending the individual information packets, the sender in a network-coded system transmits linear combinations of information packets. The recipient continues to receive these combinations until it collects enough of them to solve a system of linear equations and obtain the information packets. Network coding eliminates the need for retransmission of failed packets and makes it unnecessary to send packet-by-packet acknowledgements. As a result, overhead is reduced and throughput is increased.

In this dissertation, we investigate applications of network coding to ad hoc packet radio networks in which a file must be delivered to multiple destinations. We use the term *broadcast* to refer to the transfer of a file from one node, referred to as the *source*, to every other node in the network. We use the term *multicast* to refer to the transfer of a file from a source to a subset of the nodes in the network. Each node in this subset is called a *destination*. A broadcast network may therefore be thought of as a multicast network in which every node other than the source is a destination. A *relay* node is a node that is responsible for transferring packets between two or more of its neighbors. In a broadcast network, each relay node is also a destination node. In a multicast network, some relay nodes may not be destinations themselves. There are two possible situations for a relay node in a packet radio network. In the first situation, the relay transfers packets from one neighbor to another before obtaining the complete file itself. A relay operating in this manner must wait for new incoming packets before it can make transmissions. The other situation arises once the relay node obtains the complete file from the source. Now the relay no longer needs to wait for incoming packets; instead, it can act as an independent source node and send a continuous stream of packets to its neighbor without relying on any other nodes in the network. We call a relay node that operates in this manner an *intermediate source*.

Several methods have been proposed in the literature for network-coded broadcast and multicast file distribution in wireless networks. For example, applications of fountain coding to file distribution in cellular systems is considered in [4]. In [5], techniques for fountain-coded broadcast in half-duplex packet radio networks are described. Both [4] and [5] address scenarios in which a source transmits fountain-coded packets directly to a number of destinations; situations in which packets must be relayed by intermediate nodes are not considered. Strategies for fountain coding in a cooperative relay network are described in [6]. In [7], protocols are provided for fountain-coded cooperative communications in clustered ad hoc networks. While the methods of both [6] and [7] are applicable to half-duplex packet radios, they require nodes to receive packets from multiple senders simultaneously. Methods that employ random linear network coding for multicast in ad hoc networks are proposed in [8] and [9]. These methods require that a relay node transmit a network-coded packet every time it receives a new network-coded packet from the source. Having a half-duplex relay node transmit packets before it has decoded the file increases decoding-completion time at the relay node. Such delays may be undesirable when the relay node is also a destination. Also, the methods in [8] and [9] do not take advantage of the fact that once a relay node has obtained the file, it can stop receiving packets from the source and begin acting as an intermediate source for its neighbors. In addition, the methods require that the expected transmission count (ETX) metric be available for each link in the network. The selection of relay nodes and the forwarding decisions made by relay nodes are based entirely on the ETX metric.

In the methods we suggest, nodes are not required to have full-duplex capabilities, a sender needs to transmit only one packet at a time, and a recipient has to receive packets from only one sender at a time. We illustrated some aspects of our approach in [10]–[15] by considering broadcast and multicast file transfers in small networks. Here we provide general frameworks that apply to networks of arbitrary size and topology. Our techniques exploit the fact that a transmission made by a sender in a wireless network may reach not only its intended recipients but also some of the other nodes in the vicinity. If a node sends a packet to another node and a third node receives it, then we say that the third node *overheard* the packet. We show that proper application of network coding permits the nodes to benefit from overheard transmissions. In our methods, multiple relay nodes can cooperate to transfer a file to one or more destination nodes without resorting to complicated mechanisms for dealing with duplicate packets. A relay node that is also a destination is not required to transmit packets before it has obtained the complete file. Our methods do not depend on the availability of any specific link metric.

To obtain high throughput in any communication network, it is necessary that a low probability of packet erasure be maintained on the links while avoiding unnecessary reductions in the information rate of the transmissions. This can be achieved by choosing a suitable modulation format and channel-coding rate for each link. The appropriate choice of the modulation and channel code depends on the signal-to-noise ratio (SNR) of the link. A code-modulation combination with high information rate is suitable for a link with high SNR, whereas a low-rate combination should be used if the SNR of the link is low. Unfortunately, the SNR of a link in a typical wireless network does not remain constant throughout the file transfer; rather, it varies with time due to such phenomena as fading and shadowing. Consequently, no single code-modulation combination is suitable for all transmissions over a link. To remedy this situation, our approach includes an adaptive transmission protocol that permits senders to adapt the modulation format and channel code from one packet to the next in response to changes in link conditions. The control information for the protocol is a simple receiver statistic computed at the recipients.

We discuss three classes of file-transfer strategies in this dissertation: We first describe a set of techniques that can be used in any broadcast network. Then we examine a special type of multicast network in which a source tries to transfer a file to a cluster of remotely situated destinations with the help of two relay nodes that do not need the file themselves. Finally, we present strategies for multicast file distribution in networks with arbitrary topologies.

We employ bit-level simulations for numerical evaluations of file-transfer strategies in networks with fading on the links. We show that each of our proposed methods performs significantly better than techniques that rely on ARQ. We also provide analytical upper bounds on the throughput of file transfers in networks comprising four nodes. While exact mathematical analyses of large networks are typically very complicated or even intractable, performance bounds obtained for smaller networks can provide useful insights into the effectiveness of different approaches to file distribution. We demonstrate that our network-coded file-distribution techniques are able to achieve throughput that is very close to the upper bounds.

Chapter 2

Network Coding

Consider the transfer of a file from a source to a destination over a wireless link. The file consists of K *information packets*, which are fixed-length information-bit sequences. Let the set of information packets be denoted by $\{\mathbf{s}_i : 1 \leq i \leq K\}$. The network encoding process is depicted in Figure 2.1. The information packets are the input symbols to the network encoder, which can generate a potentially infinite stream of *network-coded packets* (NC packets) by making random linear combinations of its inputs. An NC packet produced by the encoder can be expressed as

$$\mathbf{b} = \sum_{j=1}^{K} a_j \mathbf{s}_j, \tag{2.1}$$

where the *encoding coefficients* $\{a_j : 1 \leq j \leq K\}$ are chosen randomly or pseudorandomly from a finite field. The sequence $(a_1, a_2, \ldots, a_K)$ is referred to as the *encoding vector* for the NC packet.

We restrict attention to network coding in GF(2). For such encoders, each encoding coefficient is either 0 or 1 and the NC packets can be formed by performing bitwise XOR of all information packets that have coefficient 1. Each NC packet is encoded by a channel

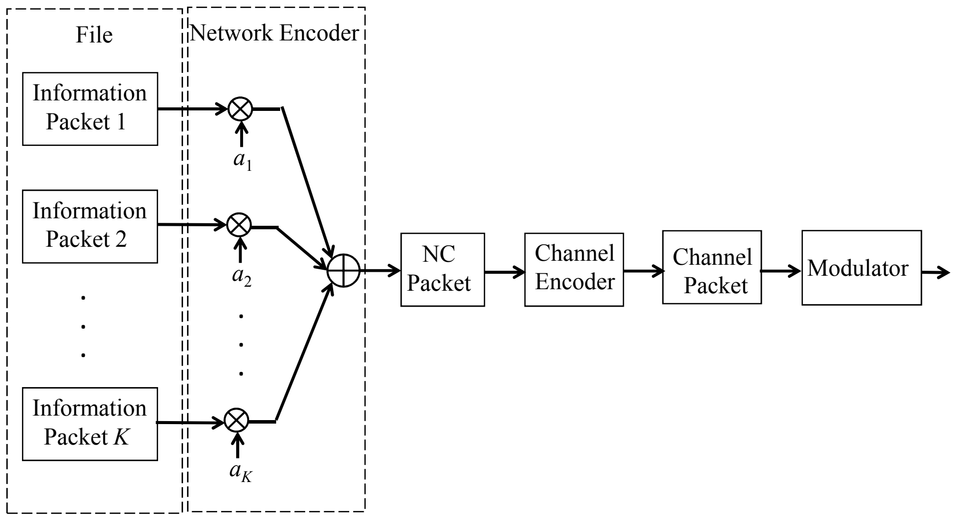


Figure 2.1: Block diagram of the network encoder, channel encoder, and modulator.

code to form a *channel packet*. The channel packet is then modulated and transmitted over the channel. At the recipient, the received signal is demodulated and the channel code is decoded by a channel decoder. The underlying NC packet is said to have been *recovered* by the recipient if the decoding of the channel code is successful; otherwise, the packet is *erased*. The recipient maintains a *decoding matrix*, the rows of which are the encoding vectors for the NC packets that the recipient has recovered. Each recovered NC packet adds a new row to this matrix. An NC packet is *innovative* if it increases the rank of the decoding matrix, which happens when the encoding vector for the NC packet is linearly independent of the encoding vectors for the NC packets previously recovered by the recipient. Once a recipient recovers K innovative NC packets, thus forming a decoding matrix of full rank, it can compute all K information packets in the file by solving a system of linear equations. We say that the file has been *decoded* when all information packets have been obtained.

Because not every incoming NC packet may be innovative, the number of NC packets that must be recovered in order to decode the file may exceed K. Suppose that $K+\varepsilon$

is the number of NC packets that had to be recovered by a recipient before it was able to decode the file. We refer to ε as the number of *excess packets*. For a well-designed network code, the expected number of excess packets is typically quite small.

Because *any* set of K innovative packets suffices for decoding the file, there is no need for the sender to retransmit erased NC packets. Instead, it can continue to send new NC packets at each transmission opportunity. Consequently, it is not necessary for the recipient to acknowledge the recovery of individual NC packets or for the sender to keep track of which NC packets were erased.

The fundamental difference between fountain coding and random linear network coding lies in how the encoding coefficients a_j in (2.1) are chosen. A simple encoding model that results in a uniform degree distribution[1] on the NC packets can be used for RLNC over GF(2). It this model, for each NC packet, the encoder chooses a degree d' at random according to a uniform distribution on the set of integers $\{1, 2, \ldots, K\}$. Next, the encoder selects d' coefficients at random, sets them to 1, sets the rest to 0, and applies (2.1). This is equivalent to selecting d' information packets at random and computing the bitwise XOR of the selected packets to obtain the NC packet. At the recipient, Gaussian elimination is applied to the decoding matrix in order to obtain the information packets.

For fountain coding, the encoding coefficients are chosen such that the degree distribution of the NC packets satisfy certain criteria. For example, a class of fountain codes known as Luby Transform codes [16] require that the degrees of the NC packets have the robust soliton distribution. By using special degree distributions, fountain codes permit recipients to use decoding algorithms that have significantly lower computational complexity than Gaussian elimination [17]. Rather than locally generating the encoding coefficients, all senders in a fountain-coded system use the same mapping between the sequence num-

[1]The degree of an NC packet is the number of information packets that were combined to form that NC packet.

ber of an NC packet and its encoding vector. This mapping is known to all nodes in the network.

To reduce the decoding complexity of RLNC, the K information packets in the file may be divided into g disjoint *generations* of d packets each. For each NC packet, the RLNC encoder first chooses a generation, selects a degree d' at random according to a uniform distribution on the set of integers $\{1,2,\ldots,d\}$, and then combines d' information packets at random from the chosen generation. The generation for each NC packet may be chosen in several different ways. For example, the sender may continue sending NC packets from a generation until the generation is decoded by its recipients and then proceed to the next generation. Alternatively, the generation for each transmission may be chosen at random according to a uniform distribution on the set of integers $\{1,2,\ldots,g\}$. It is also possible to select generations in a round-robin fashion, sending a block of NC packets from a generation before proceeding to the next generation. We will examine these approaches later in this dissertation.

In order for the recipient to form the decoding matrix, it must know the encoding vector for each NC packet. For RLNC, this information must be conveyed to the recipient along with the NC packets, which requires $\lceil d + \log_2 g \rceil$ bits to be appended to each NC packet. For fountain coding, because the sender and the recipient both know the mapping between the sequence numbers of the NC packets and their encoding vectors, it suffices for the sender to include only the sequence number in the NC packet's header.

Once the recipient is able to decode the file, it notifies the sender by means of an acknowledgement packet and the sender stops transmitting NC packets. If RLNC is employed, the recipient also acknowledges the decoding of each generation. The sender avoids sending NC packets from generations that have already been decoded.

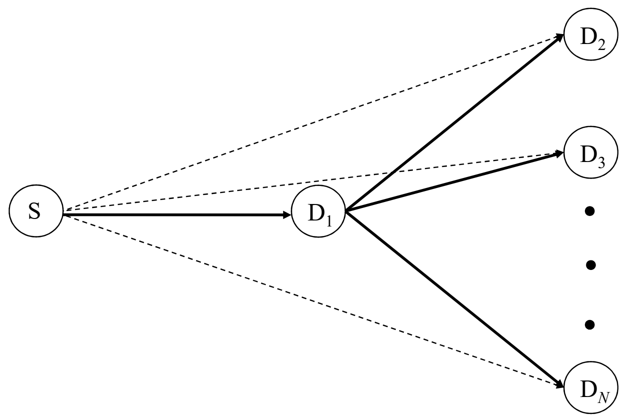


Figure 2.2: An example of a broadcast network.

2.1 Benefits of Network Coding Over ARQ

To see some of the benefits that network coding provides over ARQ, consider the network of Figure 2.2. Suppose that node S wishes to send a file consisting of K information packets to nodes D_1 through D_N using ARQ. All links in the network can support communications, but each link from S to the remote destinations (D_2 through D_N) has a much lower SNR than the link form S to D_1 and the link from D_1 to each remote destination. For this network, a typical routing protocol is likely to designate D_1 as the relay node between S and the remote destinations. Node S first delivers all information packets in the file to D_1, and then D_1 relays the packets to the remote destinations. When S transmits, only D_1 attempts to demodulate and decode the channel packets. The remote nodes ignore all transmissions from S, even though the incoming links from S may allow the nodes to overhear some packets. This choice is motivated by the fact that, for ARQ-based file distributions, it is very difficult to take advantage of overheard information packets. To benefit from overhearing, node D_1 must ensure that it does not forward those information packets that each of the remote destinations has already received from S. To that end, before D_1

begins transmissions, each remote destination must provide D_1 with a list of the sequence numbers of the information packets that the destination was able to overhear from S. This requires the exchange of a large amount of control information between D_1 and the remote nodes. The amount of control information grows with N and K. Furthermore, the throughput benefits of overhearing may not always be significant. For example, if $N-1$ disjoint sets of information packets are erased at the $N-1$ remote destinations during the destinations' attempts to overhear from S, each packet sent by D_1 is of interest to only one node and it is a duplicate of a previously received packet for each of the remaining $N-2$ nodes.

But if ARQ is replaced by network coding, a node can decode the file after it has received any set of K innovative NC packets. This property of network coding can be utilized to devise strategies in which the remote nodes overhear NC packets from S until D_1 decodes the file, and when D_1 begins transmissions, it continues the network coding process from where S left off. As we will show in Chapter 4, fountain coding can be used in this scheme in a way that duplicate NC packets are never sent by D_1 and RLNC can be used to ensure a negligibly small probability of duplicate packets. Also, in the unlikely event that a duplicate NC packet is generated, it does not lead to a duplicate information packet at the network decoder's output. Therefore, mechanisms for duplicate detection need not be employed at the receivers when network coding is used.

The benefits of network coding are not limited to exploiting the broadcast nature of the wireless medium. Network coding also provides flexibility in how intermediate nodes are chosen for the file transfer. For example, suppose that node D_2 in Figure 2.2 is the first among the remote nodes to decode the file. Also suppose that D_2 has better links to the remaining remote nodes than does D_1. In a network-coded system, D_2 can instruct D_1 to cease transmissions, apply network coding to the file, and start transmitting packets to the remaining remote destinations without having to first learn which packets were previously delivered to the destinations by S and D_1.

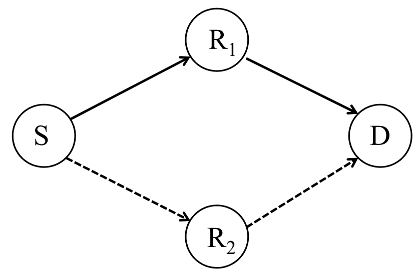


Figure 2.3: A four-node relay network.

Network coding can also exploit the presence of nodes that are neither destinations nor designated relays in the conventional sense. For example, consider the network in Figure 2.3. Suppose that source S wishes to send a file consisting of K information packets to destination D but the SNR of the direct link from S to D is too poor to permit any communication. Nodes R_1 and R_2, neither of which needs the file but are available to serve as relays, have usable incoming links from S and outgoing links to D. Suppose that the incoming and outgoing links for node R_1 have higher SNR than the corresponding links for node R_2. In this network, traditional ARQ-based approaches are likely to transfer the file from S to D by way of R_1 alone. In principle, R_2 may be allowed to forward packets from S to D until R_1 obtains the file, but it is rather complicated to do so due to the problem of duplicate packets as discussed earlier. However, with the use of network coding, it is simple to implement schemes in which R_2 forwards packet from S to D until R_1 decodes the file. When R_1 is ready to transmit, it instructs S and R_2 to cease transmissions, continues the network encoding process from where S had left off, and delivers a stream of packets to D, none of which is a duplicate of what D had previously received from R_2.

In yet another possibility, both R_1 and R_2 in the network of Figure 2.3 can take turns to receive NC packets from S and send NC packets to D. Node R_1 receives from S until it recovers a certain number of NC packets. Then R_1 begins sending those packets to D while

R_2 begins receiving packets from S. After sending all packets in its buffer, R_1 returns to its receive mode while R_2 switches to its transmit mode and sends to D the recently recovered packets. The relay nodes continue to alternate between receiving and transmitting until either D decodes the file or one of the relay nodes decodes it. In the latter scenario, the relay node that has decoded the file instructs S and the other relay to stop transmitting, continues the network encoding process from where S had left off, and keeps sending NC packets to D until the latter decodes the file. Again, the use of network coding ensures that a relay node does not have to take into account the identities of the packets that the other relay sent to D, which greatly simplifies the implementation of the scheme.

Chapter 3

System Model and Performance Metrics

We consider packet radio networks in which (a) all nodes are half-duplex, (b) nodes are not required to transmit multiple packets simultaneously, (c) nodes are not required to receive packets from multiple senders simultaneously, (d) the source's transmission may not reach all nodes in the network, so some form of relaying may be required, and (e) the transmission method used by the radios permits multiple radios to send packets simultaneously with negligible mutual interference in the radio receivers. For (e), the radios might use frequency-division multiple access or some form of spread-spectrum multiple access.

3.1 Definitions

We say that a *session* has begun when the source starts transmitting packets from the file. The session ends as soon as all nodes in the network stop sending packets. A session *succeeds* if each of the destination nodes is able to obtain a copy of the file by the end of the session. In rare occasions, a destination node may not have operational incoming links from any of its neighbors. In such situations, the neighbor responsible for transferring the file to the disadvantaged node stops sending packets after making several unsuccessful attempts to deliver packets. We say that a session has *failed* if there is at least one destination node in the network that did not receive the file by the end of the session. The *completion time* for a destination is the time that elapses since the beginning of the session until the destination obtains the file. The *session completion time* is the time that elapses since the beginning of the session until its end.

We use the term *forwarding* to refer to a relay node's action of transmitting a previously-received packet without combining it with any other packets. When the term forwarding is not used, it must be assumed that the transmitted packet was generated by some network-coding operation performed at the transmitting node.

3.2 Simulation of Network Coding

We denote the number of information packets in the file by K. Unless stated to the contrary, the numerical results presented in this dissertation are for information packets consisting of 2400 bits.

The systematic raptor code described in [18] is used for fountain coding. To reduce simulation run times, we employ the probabilistic formulation of [4] to simulate the decoding of the raptor code. Because the code is systematic, the recipient can decode the file with probability 1 if it is able to recover the set of K fountain-coded packets with sequence numbers 0 through $K-1$. In all other cases, the probability that fountain decoding succeeds upon the recovery of the ith fountain-coded packet ($i \geq K$) after having failed for each of the previously recovered packets can be approximated by

$$P_s[i,K] \approx \begin{cases} 0.15, & i = K, \\ 0.433, & i > K, \end{cases} \tag{3.1}$$

More details on this approximation are given in Appendix B.

For our performance results on RLNC, the file is divided into 5 generations of 100 packets each. Gaussian elimination is used for decoding the generations.

3.3 Channel Model

We consider networks with correlated block fading on each link. The fading is independent from link to link. The fade level on a link remains constant over a channel packet but may change from one packet to the next. The fading is simulated using finite-state Markov-chain models of the Nakagami-m fading process [19], the parameters for which are derived according to the method given in [20]. The Markov chains are assumed to be operating in the steady state. Each Markov chain has 12 states and each state corresponds to a unique fade level.

Channel transitions allowed by these Markov-chain models are not restricted to adjacent states alone. The transition probability is a function of the normalized Doppler frequency $f_d T_s$, where f_d is the Doppler frequency of the channel and T_s is the average time duration between the start of one packet transmission to the start of the next. The correlation coefficient for samples of the Nakagami-m fading process that are separated in time by T_s is given by $\rho = J_0^2(2\pi f_d T_s)$, where J_0 is the Bessel function of the first kind of order zero.

We provide performance results for $m = 1$, 2.5, and 3.25. A smaller value of m indicates more severe fading. In particular, $m = 1$ represents Rayleigh fading, which is the most severe fading typically encountered on a wireless link. The fading processes for $m = 2.5$ and $m = 3.25$ are approximately Rician with a specular-to-diffuse ratio of 3.4 and 5, respectively [21], [22]. The channel gain in dB when the Markov chain is in state j is given by $G_j = j\xi - \xi_0$ dB for $0 \leq j \leq 11$, where ξ and ξ_0 depend on m. The value of ξ is 2, 1.25, and 1 for $m = 1$, $m = 2.5$, and $m = 3.25$, respectively. The corresponding values of

ξ_0 are 16, 9, and 7. The normalized Doppler frequency in our simulations is set to 0.02, which results in relatively fast fading channels.

3.4 Modulation Formats and Channel Codes

The transceivers at the nodes are capable of generating and demodulating signals with biorthogonal key (BOK), phase-shift key (PSK), and quadrature-amplitude modulation (QAM). Specifically, we consider four modulation formats: 64-biorthogonal key (64-BOK), binary phase-shift key (BPSK), quadriphase shift key (QPSK), and 16-quadrature amplitude modulation (16-QAM). The elemental rectangular pulses that constitute a modulation symbol are referred to as *modulation chips*. While BPSK, QPSK, and 16-QAM have one modulation chip per modulation symbol, a 64-BOK modulation symbol consists of 32 modulation chips.

Bit-interleaved coded modulation is used for each transmitted packet. The nodes are equipped with encoders and decoders for five turbo product codes [23] of rates 0.260, 0.346, 0.472, 0.620, and 0.766 with block lengths of 4608, 6930, 5082, 3872, and 3135, respectively. The channel code of rate 0.260 has 1200 information bits per codeword; therefore, each channel packet consists of two codewords when this code is used. The remaining four channel codes have 2400 information bits per codeword; so the channel packets consist of one codeword each.

A sender can use a fixed modulation format and channel code for all packets or it can adapt the code-modulation combination in response to changes in channel conditions with the help of an adaptive modulation and channel coding (AMCC) protocol. When AMCC is used, each recipient calculates a statistic called the *error count* for the recovered packets. The error count is the number of bit errors observed at the demodulator output prior to the decoding of the channel code. An interval test is applied to the error count to determine the code-modulation combination that the recipient would like for the sender to use for the next packet. The recipient's choice is conveyed to the sender by means of a feedback packet. A detailed description of the AMCC protocol is given in Appendix C.

3.5 Measure of SNR

To avoid increasing interference at unintended recipients, a sender keeps the bandwidth of the transmitted signal and its average power constant. This is accomplished by keeping the duration and energy of a modulation chip the same for all modulation formats. Therefore, the *chip-energy-to-noise-density ratio* (CENR) provides an appropriate measure of the SNR on a link. The value of CENR in dB is given by $\text{CENR} = 10\log_{10}(E_c/N_0)$, where E_c is the average energy per modulation chip and N_0 is the one-sided power-spectral density of the additive Gaussian noise on the link. We refer to the CENR of a link in the absence of fading as the *nominal* CENR. The actual value of CENR for a packet is the nominal CENR plus the fade level G_j at the time the packet is transmitted.

3.6 Performance Metrics

We employ two performance metrics, one based on the session completion time and the other on the individual completion times for the destination nodes. Unless stated otherwise, one unit of time is defined as the duration of one modulation chip.

3.6.1 Session Throughput

The session throughput metric is based on the session completion time. Let L_t be the total number of sessions that were simulated and let L_s be the total number of sessions that were successful. Denote by T_i the duration of the ith session and let b be the number of information bits carried by each channel packet. The session throughput is defined as

$$\bar{S} = \frac{L_s K b}{\sum_{i=1}^{L_t} T_i}. \tag{3.2}$$

3.6.2 Mean Destination Throughput

In a given session, different nodes in the network may take different amounts of time to obtain the file. Because session throughput takes into account the duration of the entire session, it reflects the completion time of the node that is the last to receive the file. As a measure of the individual completion times of the nodes, we use the metric mean destination throughput. It is defined as

$$\bar{S}_d = \frac{Kb \sum_{i=1}^{L_t} |\Omega_i|}{\sum_{i=1}^{L_t} \sum_{j \in \Omega_i} \tilde{T}_{i,j}}, \tag{3.3}$$

where $\tilde{T}_{i,j}$ is the completion time of destination j for the ith session, Ω_i is the set of destination nodes that were able to obtain the file by the end of the ith session, and $|\Omega_i|$ denotes the cardinality of Ω_i.

Chapter 4

Broadcast Networks

Applications of fountain coding and random linear network coding to the reliable transfer of a file from a source to all other nodes in a half-duplex packet radio network are discussed in this chapter. We describe four modes of network-coded broadcast.

Our techniques employ a spanning tree [24] rooted at the source node. Each node in a spanning tree has exactly one parent, the only exception being the root node, which has none. A parent has the responsibility of transferring packets from the file to its children. We refer to a parent as the *primary relay* for its children. Figure 4.1 illustrates a spanning tree in an ad hoc network. The spanning tree consists of the links depicted by solid lines. Node S is the source; therefore, it is the root node of the spanning tree. Nodes N_5 through N_8 are the primary relays for four disjoint sets of leaf nodes. We define the number of *hops* between two nodes as the number of edges in the spanning tree that must be traversed in order to travel from one node to the other. For example, nodes N_1 through N_8 are one hop away from node S whereas all other nodes are two hops away from S.

In conventional methods for broadcast using a spanning tree, each recipient receives packets only from its parent on the spanning tree (i.e., its primary relay). In our methods, on the other hand, a node may receive packets not only from its primary relay, but also

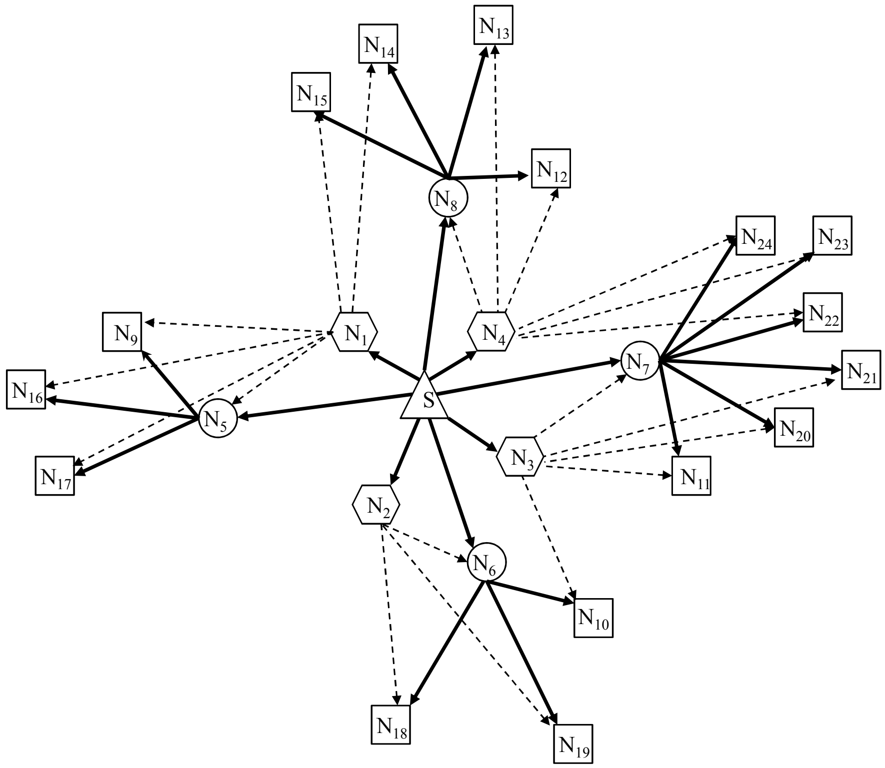


Figure 4.1: A broadcast network with a spanning tree (solid lines) and secondary links (dashed lines).

from other senders. In some modes of our network-coded broadcast, a destination may be provided with a *secondary relay*, which is the destination's closest neighbor on an alternate path between the source and the destination. In Figure 4.1, the links between the nodes and their respective secondary relays are shown as dashed lines. For example, N_1 is the secondary relay for N_5, N_9, N_{16}, and N_{17}.

We call N_i an *upstream node* for N_j if N_i lies on the path from the source to N_j in the spanning tree. Node N_j is called a *downstream node* for N_i if the latter is an upstream node for the former. We define a *primary incoming link* for a node as the edge in the spanning tree that connects the node and its primary relay. The *secondary incoming link* for a node is the link between its secondary relay and the node itself.

A variety of methods can be used to assign the relays. The primary relays can be obtained by using one of many tree-construction algorithms (e.g., [24]). It only requires minor modifications to the algorithms in order to assign the secondary relays. One strategy for selecting relay nodes is described in Appendix D.

4.1 Modes of Operation

The four modes of network-coded broadcast differ in how they utilize the secondary relay nodes. The modes are named as follows:

- Mode NS: ("No Secondary") Does not utilize secondary relay nodes.
- Mode TS: ("Temporary Secondary") Uses secondary relays as temporary intermediate sources.
- Mode CS: ("Choose Secondary") Permits recipients to choose between primary and secondary relays.
- Mode AS: ("All Secondary") Any node in the network can become a secondary relay and recipients are permitted to choose between primary and secondary relays.

The description below applies to both fountain coding and random linear network coding. Measures specific to fountain-coded broadcast will be described later. In the following, the term *packet* refers to an NC packet unless stated otherwise.

Mode NS

Mode NS is the simplest mode of operation for network-coded broadcast. It employs the spanning tree but does not utilize the secondary relays. The source begins the session by applying network coding to the file and sending NC packets to its children. Once a recipient recovers K innovative NC packets, it performs network decoding to obtain the information packets. If the recipient is a primary relay node, then the recipient broadcasts a control packet to its children informing them that it has decoded the file and is ready to send NC packets. Each child responds to this announcement with a reply packet. The relay node then applies network coding to the information packets and starts sending NC packets to its children. Recipients are not required to acknowledge the receipt of the NC packets, although feedback packets may be sent for other purposes (e.g., to provide control information for adaptive transmission). However, the decoding of the file is acknowledged by each recipient. The recipients also acknowledge the decoding of each generation if RLNC is used. Unlike conventional file transfers, each node tries to overhear channel packets transmitted by upstream nodes that are not its primary relay. If multiple upstream nodes are transmitting, then the recipient listens to the node that is the fewest hops away. For example, node N_{16} in Figure 4.1 tries to overhear the channel packets transmitted by S until N_5 obtains the file and starts transmitting. If both S and N_5 are transmitting, then N_{16} listens to N_5's transmissions because N_5 is one hop away from it as opposed to S which is two hops away.

Mode TS

Mode TS uses the spanning tree and allows nodes to overhear packets from upstream neighbors in a manner identical to mode NS. However, mode TS also utilizes the secondary relay nodes as temporary intermediate sources. When a secondary relay decodes the file, it broadcasts a control packet offering to send NC packets to its children. A child accepts this offer only if its primary relay has not started sending packets yet. If a secondary relay has at least one child that is willing to receive packets, it applies network coding to the information packets and starts sending NC packets. The child continues to receive packets from the secondary relay until its primary relay decodes the file. Once its primary relay starts transmitting, the child stops receiving packets form the secondary relay and becomes a recipient of the primary relay. A secondary relay stops transmitting when all of its children have begun receiving packets from their respective primary relays or have decoded the file.

For example, in the network of Figure 4.1, suppose that node N_1 decodes the file before N_5 and N_8 are able to do so. In mode TS, N_1 broadcasts a control packet offering to send NC packets to its children N_9, N_{14}, N_{15}, N_{16}, and N_{17}. The children accept N_1's offer because their primary relay nodes have not yet decoded the file. Therefore, N_1 applies network coding to the file and starts transmitting packets. After a period of time, suppose that N_5 decodes the file while N_8 is still receiving packets from S. At this point, N_9, N_{16}, and N_{17} stop receiving packets from N_1 and become recipients of N_5. Node N_1 continues to send NC packets to N_{14} and N_{15} and stops only when N_8 decodes the file and starts transmitting.

Mode CS

Mode CS operates in a manner similar to mode TS, but a node accepts its secondary relay's offer to send NC packets if either of the following two criteria is met: (a) the node's primary relay has not started sending NC packets yet, or (b) the long-term condition of its secondary incoming link is better than the long-term condition of its primary incoming link.

Notice that evaluation of the second criterion only requires estimates of the *long-term* or *average* condition of incoming links and does not require any measurements of instantaneous link quality. Because it is common for nodes in an ad hoc network to periodically exchange control messages, the nodes often have information about the long-term conditions of the incoming links. Any suitable metric can be used as a measure of long-term link quality, including estimates of the nominal SNR, bit-error or packet-error rates observed over a period of time, etc.. Also, exact values of long-term link qualities are not required; all that is needed is an estimate of the *relative* qualities of the incoming links. For our performance results, we assume that when a destination in mode CS has to choose between two potential intermediate sources, the destination chooses the one from which it has the incoming link with the higher nominal CENR.

For example, suppose that node N_2 in Figure 4.1 decodes the file before N_6 is able to do so. Because N_2 is a secondary relay node for N_6, N_{18}, and N_{19}, node N_2 broadcasts a control packet offering to begin transmissions. Also suppose that the secondary incoming link for N_6 has higher nominal CENR than its primary incoming link. In this situation, mode CS requires N_6 to stop receiving from S and become a recipient of N_2. Also, N_{18} and N_{19} start receiving packets from N_6 because their primary relay N_6 has not started transmitting yet. If the secondary incoming link N_2–N_{18} has higher nominal CENR than the primary incoming link N_6–N_{18}, then mode CS allows N_{18} to decline the offer N_6 makes when it eventually decodes the file.

Mode AS

In mode AS, any node that has decoded the file is eligible to become an intermediate source. Consequently, no secondary relays are assigned beforehand in this mode. Nodes that are more than one hop away from the source begin their participation in the session by attempting to overhear the transmissions of their upstream nodes. Suppose that a recipient that has not yet decoded the file receives a control packet from a node offering to become an intermediate source. The offer is accepted if (a) the recipient is currently not receiving packets from any other sender or (b) the incoming link over which the recipient is currently receiving packets is inferior to the incoming link from the new intermediate source. As in mode CS, recipients use the nominal CENR of incoming links as a measure of their long-term condition.

In each of the modes, senders may adapt the modulation format and channel-coding rate from packet to packet with the help of the AMCC protocol described in Appendix C. When AMCC is used, only the nodes that are intended recipients of a transmission send feedback packets containing their suggested indexes to the sender; no feedback is sent in response to overheard packets. For example, when node S in Figure 4.1 sends packets, nodes N_1 through N_8 provide feedback whereas nodes N_9 through N_{24} do not send any feedback even though they may overhear some packets.

4.2 Duplicate NC Packets in Network-Coded Broadcast

Because the encoding vector for each RLNC packet is chosen uniformly at random, the probability of the encoder generating duplicate NC packets by choosing the same encoding vector more than once is very low for any moderate to large generation size. In the unlikely event that a duplicate NC packet is generated, it is treated by the recipient's network decoder as just another non-innovative NC packet, which adds a linearly dependent row to the decoding matrix. Such rows are eliminated during Gaussian elimination. Upon completion of network decoding, the output of the decoder is the set of K information packets; in other words, duplicate NC packets at the input to the network decoder do not result in duplicate information packets at the decoder's output.

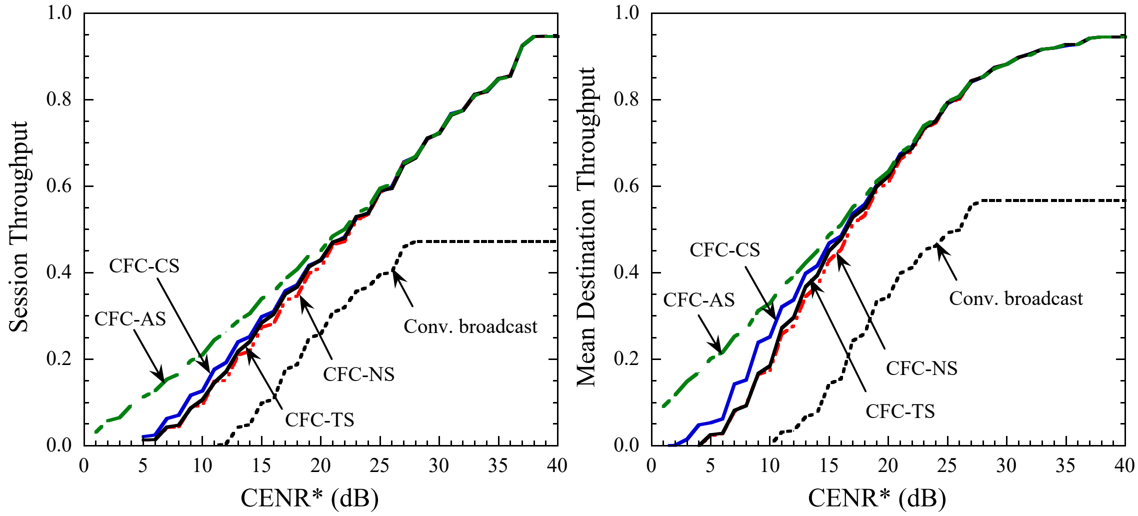
Our fountain-coded broadcast schemes, on the other hand, prevent senders from generating duplicate NC packets by using a strategy referred to as *continued fountain coding* (CFC). Each fountain-coded packet carries with it a sequence number, which uniquely defines the encoding vector for the packet. We refer to the sequence number of the NC packet to be produced next by a fountain encoder as the *state* of the encoder. Recall that each sender broadcasts a control packet to potential recipients before it begins transmitting NC packets and each node that is willing to become a recipient responds with a reply packet. When continued fountain coding is employed, the recipient includes in the reply packet the largest sequence number that it has received thus far. When the sender begins transmitting fountain-coded packets, it starts its fountain encoder from a state that is one higher than the reported sequence number. If there are multiple recipients, then the sender begins encoding from a state that is one higher than the maximum of the sequence numbers reported by the recipients. This ensures that no duplicate NC packets are sent to any of the recipients.

4.3 Performance Results

In our performance results, graphs are labeled CFC or RLNC depending on which form of network coding is used. The mode of operation is appended to the label. For example, CFC-TS refers to a broadcast session in which CFC is used in mode TS; similarly, RLNC-CS implies that RLNC is used in mode CS.

As a performance benchmark, we simulate an ARQ-based broadcast scheme, which we refer to as *conventional broadcast*. Conventional broadcast uses the same spanning tree as network-coded broadcast, but employs ARQ for retransmission of failed packets. In this scheme, a node can receive channel packets only from their respective primary relays. Secondary relays are not used and nodes do not try to overhear packets.

Our first set of performance results is for the network of Figure 4.1. We employ an offset parameter λ to assign a range of nominal signal-to-noise ratios to the links in the network. In our formulation, the nominal CENR of the links from S to N_1 through N_4 is CENR*, to N_5 through N_8 is CENR$^* - \lambda$, to N_9 through N_{12} is CENR$^* - 1.5\lambda$, and to N_{13} through N_{24} is CENR$^* - 2\lambda$, where $\lambda > 0$. The nominal CENR of the links to



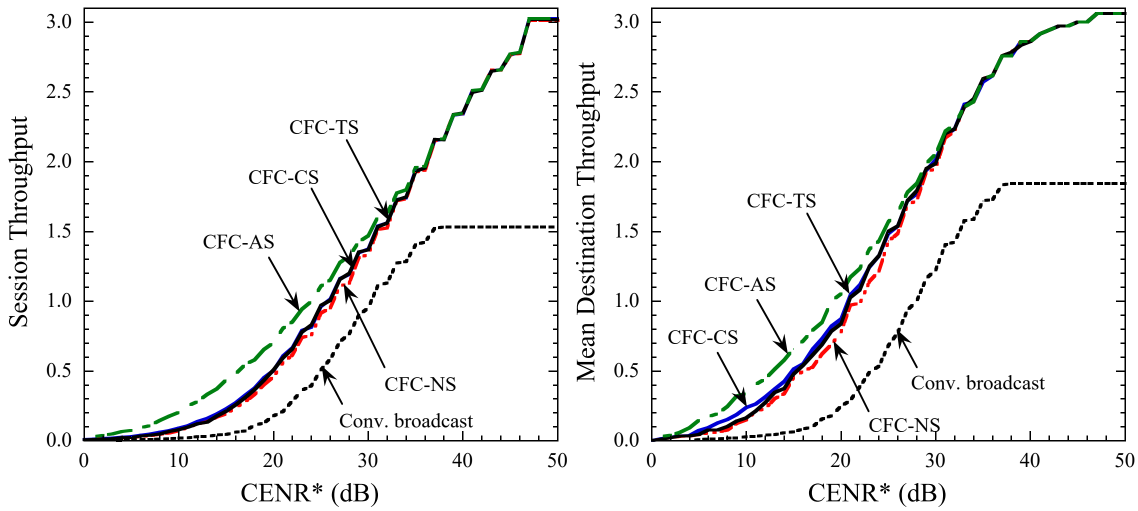
(a) Session throughput.

(b) Mean destination throughput.

Figure 4.2: Performance of different broadcast schemes in the network of Figure 4.1 when a TPC of rate 0.472 along with QPSK is used for transmissions ($\lambda = 10$ dB, $m = 1$).

N_9 through N_{12} from their respective primary relays is CENR* and from their respective secondary relays is CENR$^* - 1.5\lambda$. The nominal CENR of the links to N_{13} through N_{24} from their respective primary relays is CENR$^* - \lambda$ and from their respective secondary relays is CENR$^* - 1.8\lambda$.

The session throughput and the mean destination throughput of CFC-based broadcast and conventional broadcast in this network with Rayleigh fading (i.e., $m = 1$) on the links are shown in Figures 4.2a and 4.2b, respectively. The value of the offset parameter λ is set to 10 dB. The senders use a turbo product code (TPC) of rate 0.472 and QPSK modulation for all packets. For either performance metric, each mode of CFC significantly outperforms the conventional broadcast scheme for the entire range of CENR* shown in the figures. At high SNR, the throughput of network-coded broadcast is double that of conventional broadcast. Mode AS, by virtue of its ability to utilize all nodes in the network as potential relay nodes, provides the best performance among the CFC-based methods. Modes TS and CS provide slightly higher throughput than mode NS.



(a) Session throughput.

(b) Mean destination throughput.

Figure 4.3: Performance of different broadcast schemes in the network of Figure 4.1 when the AMCC protocol is used for transmissions ($\lambda = 10$ dB, $m = 1$).

The session throughput and the mean destination throughput for the same network when the AMCC protocol is used for transmissions are plotted in Figures 4.3a and 4.3b, respectively. We observe that each broadcast scheme gives much higher throughput with AMCC than with fixed-rate transmissions. For example, all four modes of network-coded broadcast provide a maximum throughput of 0.944 information bits per modulation chip with the fixed-rate scheme whereas their maximum throughput exceeds 3 information bits per modulation chip when AMCC is used. More comparisons between AMCC and fixed-rate transmissions will be provided later. In the performance results that follow, the AMCC protocol is used for all transmissions, unless mentioned otherwise. Also, we restrict attention to the session throughput for the remaining performance results, because we have found the mean destination throughput to display the same trends as the session throughput in each of the broadcast networks that were investigated.

Figure 4.4a shows the session throughput of CFC-based broadcast for the same network when the links have Nakagami-m fading with $m = 2.5$ and the value of λ is 6 dB.

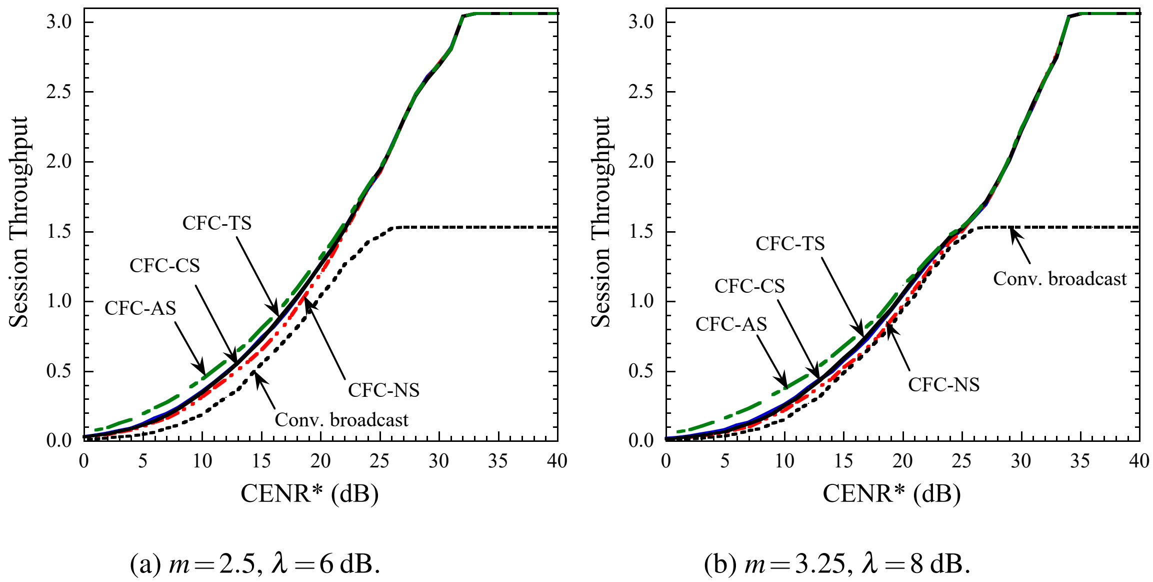


Figure 4.4: Session throughput of different broadcast schemes in the network of Figure 4.1 for two values of the Nakagami parameter m.

In Figure 4.4b, results are shown for $m = 3.25$ and $\lambda = 8$ dB. Again, mode AS outperforms modes TS and CS, which in turn perform slightly better than mode NS, and each mode of CFC significantly outperforms conventional broadcast.

Next, suppose that node N_5 moves to a new location while the other nodes remain in the same location as for Figure 4.1. The new topology is shown in Figure 4.5. The new nominal CENR of the link from N_5 to N_9 is given by $\mathrm{CENR}^* - 1.8\lambda$ and that of the links from N_5 to N_{16} and N_{17} is given by $\mathrm{CENR}^* - 2\lambda$. Clearly, for this new topology, N_1 is a better choice than N_5 to serve as the primary relay for N_9, N_{16}, and N_{17}. However, routing updates are usually costly, and are hence not performed very frequently. Therefore, we investigate a situation in which a broadcast session is conducted with the original spanning tree shown by the solid lines in Figure 4.5. The links have Rayleigh fading and the value of λ is 10 dB. The session throughput of CFC for this network is shown in Figure 4.6. The graphs show that modes CS and AS perform significantly better than modes TS and NS in this case, and each mode of CFC outperforms conventional broadcast by large margins.

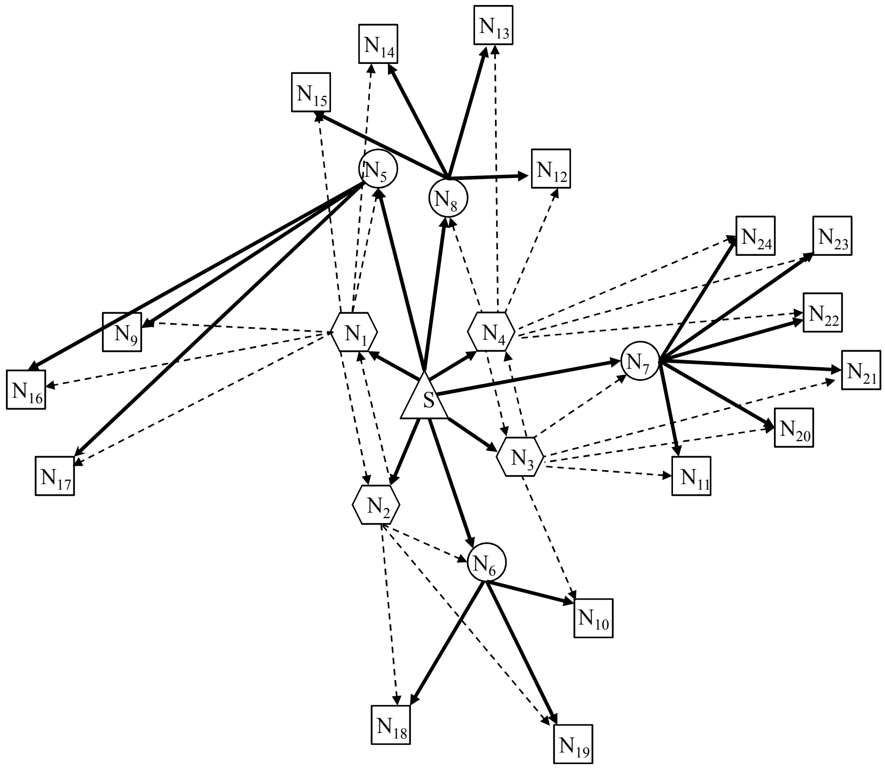


Figure 4.5: The network of Figure 4.1 after node N_5 has moved to a new location.

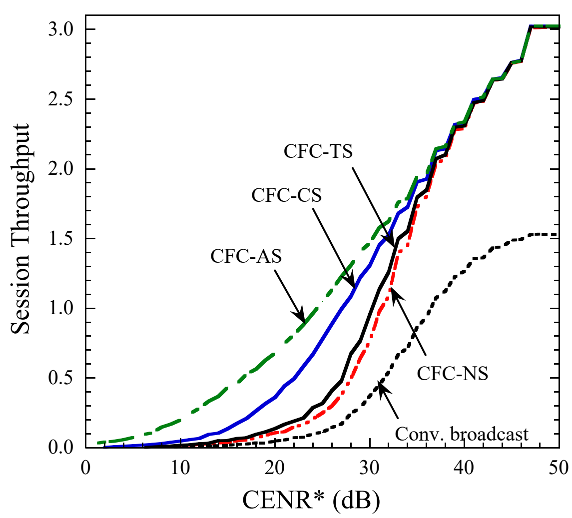


Figure 4.6: Session throughput of different broadcast schemes in the network of Figure 4.5 ($\lambda = 10$ dB, $m = 1$).

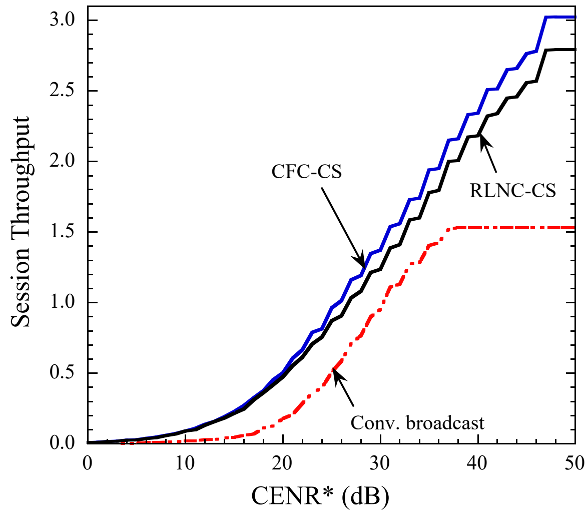
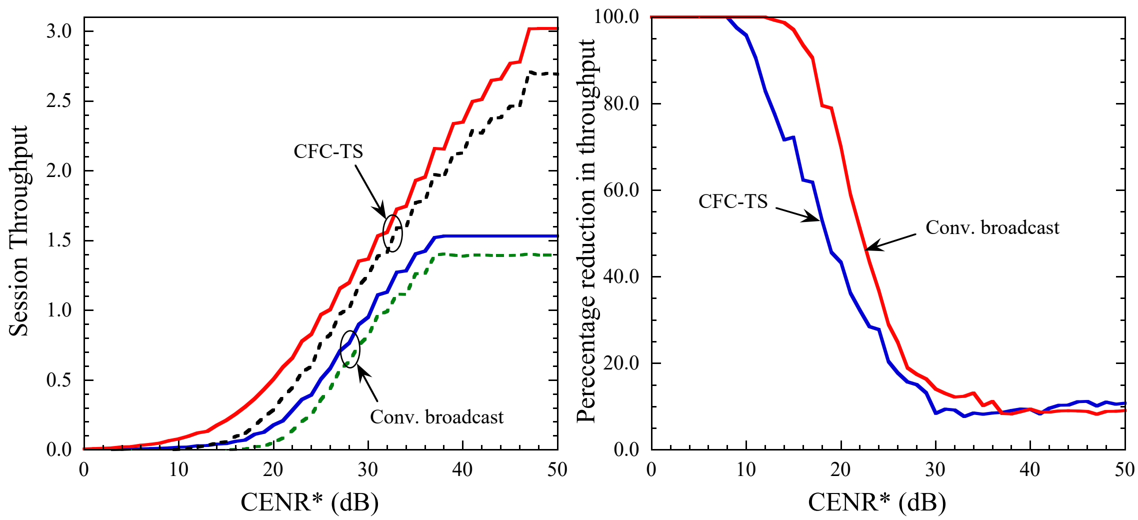


Figure 4.7: Session throughput of CFC-CS and RLNC-CS in the network of Figure 4.1 ($\lambda = 10$ dB, $m = 1$).

Figure 4.7 shows the session throughput of RLNC for the network of Figure 4.1 and compares it with that of CFC and conventional broadcast. Mode CS is used for both forms of network coding. We observe that the throughput of RLNC lags that of CFC, which is a result of the extra overhead associated with RLNC. Recall that each RLNC packet carries additional $\lceil d + \log_2 g \rceil$ bits to convey the encoding vector to the recipient, where g is the number of generations and d is the number of packets per generation. For $d = 100$ and $g = 5$, this results in an overhead of 103 bits per NC packet. Even more overhead is incurred due to the excess packets required for decoding each generation. We found that each generation requires approximately as many excess packets as required by the entire file when it is not divided into generations. However, despite its shortcomings relative to CFC, RLNC performs significantly better than conventional broadcast.

The effects of link failures in the network of Figure 4.1 are shown in Figure 4.8a. CFC-TS is used for network-coded broadcast. The relay assignments at the beginning of the session are same as those in Figure 4.1. However, during the session, links may fail randomly, prompting the affected nodes to use relay nodes other than their originally

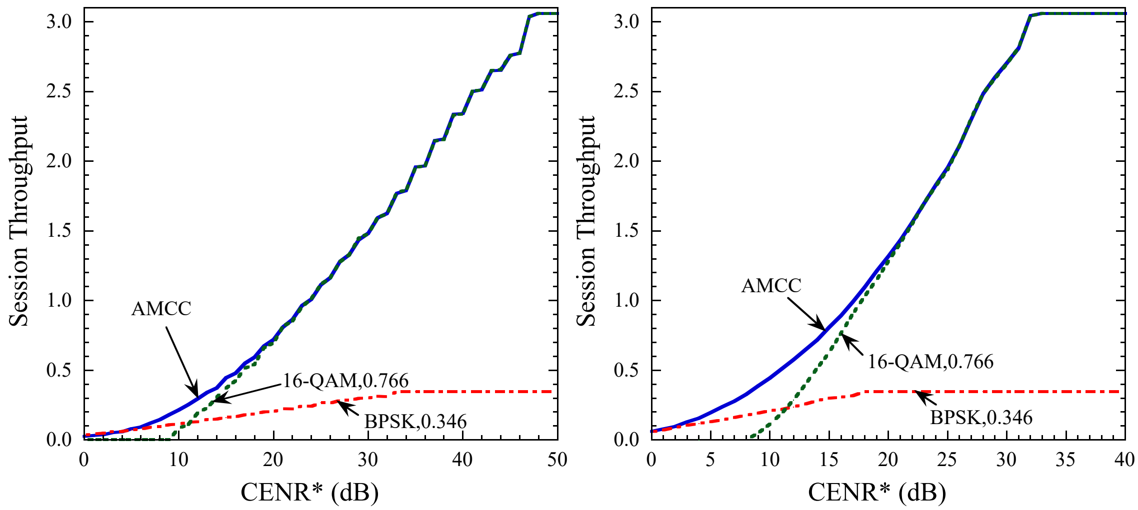


(a) Session throughput with (dashed lines) and without (solid lines) link failures.

(b) Percentage reduction in session throughput due to link failures.

Figure 4.8: Effect of link failures on the session throughput of network-coded and conventional broadcast in the network of Figure 4.1 ($\lambda = 10$ dB, $m = 1$).

assigned relays. Link failures are simulated using a two-state Markov chain with states 0 and 1. State 1 indicates that the link is operational and state 0 indicates that it is not. The state-transition probabilities are $q(0|1) = 0.0001$ and $q(1|0) = 0.0008$. The Markov chain is sampled every 3468 time units, which is the duration of a packet that uses QPSK and code rate 0.346. In our simulations, we make the simplifying assumption that when a link is broken, the nodes using that link immediately become aware of the break. When the primary incoming link to a node fails, then the link with the highest nominal CENR among the remaining links is assigned as the new primary link. If the node is not aware of any functional incoming links and it does not have the file yet, the session is assumed to have failed. As expected, both network-coded broadcast and conventional broadcast suffer some throughput degradation as a result of link failures. However, Figure 4.8b shows that for most values of CENR*, the percentage reduction in throughout due to link failures is much lower for CFC-TS than for conventional broadcast.



(a) $m = 1$, $\lambda = 10$ dB.

(b) $m = 2.5$, $\lambda = 6$ dB.

Figure 4.9: Performance of CFC-AS with and without AMCC in the network of Figure 4.1 ($\lambda = 10$ dB, $m = 1$).

In Figure 4.9, we compare the performance of CFC-AS with and without the AMCC protocol. Results are shown for two fixed-rate combinations, BPSK with code rate 0.346 and 16-QAM with code rate 0.766. The curves in Figure 4.9a are for a network in which links have Rayleigh fading and the value of λ is 10 dB. The corresponding curves for $m = 2.5$ and $\lambda = 6$ dB are shown in Figure 4.9b. Each figure demonstrates the need to adapt the modulation format and channel-coding rate to achieve high throughput over a wide range of SNR and provide some throughput at low SNR. Notice that 16-QAM with the TPC of rate 0.766 provides no throughput for CENR* below 8 dB for either set of parameters in Fig. 4.9.

Next, we consider a broadcast session in the network of Figure 4.10, which shows 14 nodes placed on a square grid. Suppose that the nominal CENR in dB of the link between two nodes in the network is given by CENR$^* = C - 10\alpha \log_{10} \tilde{\mu}$, where $\tilde{\mu}$ is the Euclidean distance between the nodes, $\alpha > 0$ is the path-loss exponent, and C is the nominal CENR that would be observed if there were no propagation loss on the link. For our numerical

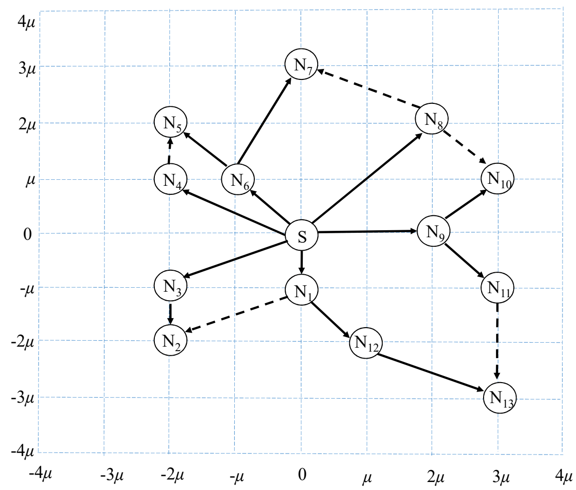
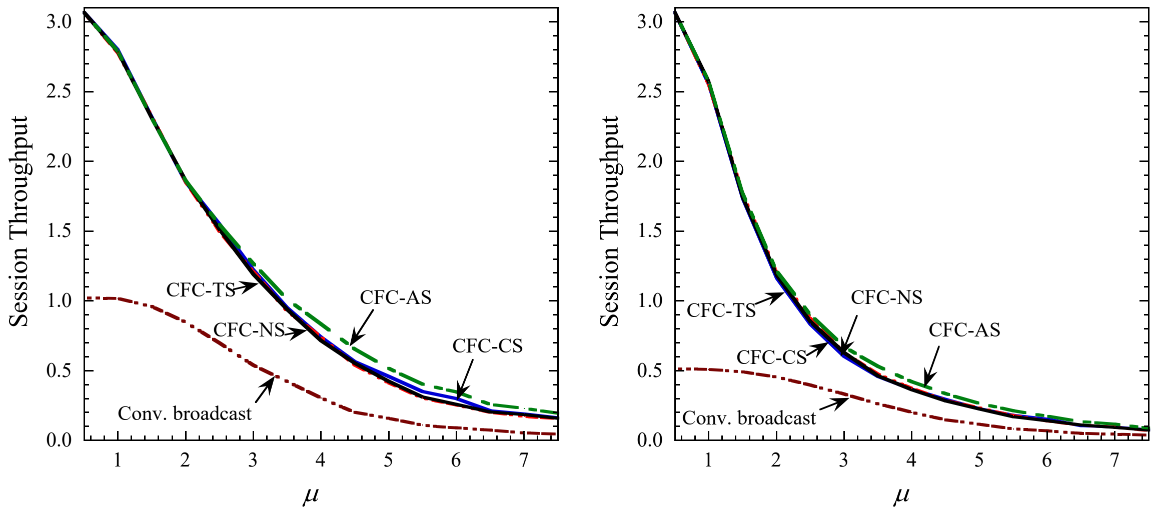


Figure 4.10: An ad hoc network with a spanning tree (solid lines) and secondary links (dashed lines).

results, we use $\alpha = 3$ and $C = 40$. Notice from the figure that the size of each constituent square is determined by μ. Therefore, a larger value of μ results in greater separation between two nodes and consequently a lower nominal CENR on the link between them.

We investigate two situations for the network of Figure 4.10, one in which nodes are allowed to transmit simultaneously and one in which they are not. For the latter scenario, we assume that the transmissions made by a node can cause interference at all other nodes within a radius of 50 units. We also assume a hypothetical channel-access protocol that allows a node to transmit only if no node within its zone of interference is currently receiving packets from another sender.

Figure 4.11a shows the session throughput of the four modes of CFC-based broadcast and that of conventional broadcast when simultaneous transmissions are permitted. The throughput is plotted as a function of μ. Each mode of CFC significantly outperforms conventional broadcast and mode AS provides the best performance, which is consistent with our observations thus far. The throughput results for the scenario in which simulta-



(a) Simultaneous transmissions permitted.

(b) Simultaneous transmissions not permitted.

Figure 4.11: Performance of network-coded broadcast and conventional broadcast in the network of Figure 4.10 with Rayleigh fading on the links.

neous transmissions are not allowed are shown in Figure 4.11b. CFC outperforms conventional broadcast by large margins in this case as well.

The performance results presented thus far do not take into account the construction of the spanning tree and the assignment of the secondary relays. As mentioned earlier, any suitable method can be used to assign the relay nodes. One method that chooses relay nodes on the basis of link resistances is described in Appendix D. The method designates a destination's closest neighbor on the least-resistance path from the source as the primary relay for the destination. The secondary relay is the closest neighbor on the path with the next higher resistance. The resistance of a link is determined by computing the bit-error rate for binary test sequences transmitted over the link.

We first apply the resistance-based relay-selection strategy to the topology of Figure 4.1 by assigning the relay nodes at the beginning of each session. The session throughput curves for CFC-based broadcast and conventional broadcast under this approach are

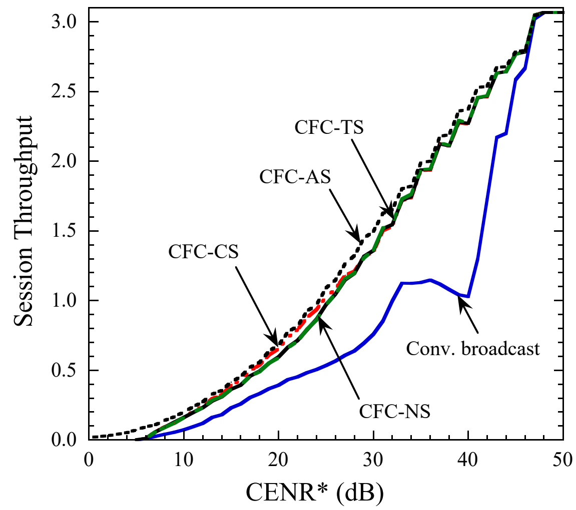


Figure 4.12: Performance of network-coded broadcast with resistance-based relay selection in the network of Figure 4.1 ($\lambda = 10$ dB, $m = 1$).

shown in Figure 4.12 for Rayleigh fading on the links and $\lambda = 10$ dB. We observe that CFC outperforms conventional broadcast by up to 10 dB. Recall that the session throughput for the same network topology and link conditions were shown in Figure 4.3a for fixed relay assignments. A major difference between Figure 4.3a and Figure 4.12 is that the high-SNR performance of conventional broadcast improves drastically when resistance-based relay selection is employed. When the SNR is high, the cost of using the direct links from the source to the remote nodes is lower than relaying packets to the remote destinations via intermediate nodes. Resistance-based relay selection takes this fact into account and instructs the source to send packets directly to all destinations at high SNR instead of employing relay nodes for the file transfer. As a result, there is a two-fold increase in the throughput of conventional broadcast compared with the fixed spanning tree of Figure 4.1. In contrast, the network-coded broadcast schemes are able to provide high throughput even with the fixed relay assignments. This is achieved by efficiently exploiting the broadcast nature of the wireless medium and, when applicable, utilizing the secondary relay nodes. We also observe that the performance gap between mode AS and the other three modes of network-

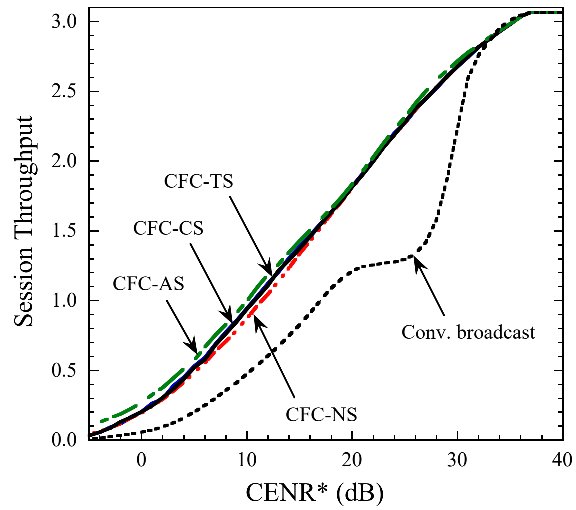


Figure 4.13: Performance of network-coded broadcast with resistance-based relay selection averaged over random link-quality assignments (15 destinations, $\lambda = 10$ dB, $m = 1$).

coded broadcast is smaller for the resistance-based relay assignments compared with the fixed relay assignments of Figure 4.1. When the relays are assigned based on link resistances, the throughput of modes NS, TS, and CS improve considerably and they perform closer to mode AS. The throughput of mode AS is the least sensitive to suboptimalities in relay assignments because this mode permits a node to transmit NC packets even when the node is not designated as a relay node.

Because the performance of any file-distribution scheme depends on the link qualities in the network, it is instructive to examine how the broadcast strategies compare when each strategy's session throughput is averaged over randomly assigned nominal CENR values for the links. To that end, we consider a network with 15 destination nodes for which the nominal CENR of each link is randomly generated prior to each simulation run. The nodes are numbered N_0 through N_{15} and node N_0 is designated as the source. The nominal CENR of the link between node N_i and node N_j is given by $\text{CENR}^* + \Lambda_{i,j}$ for $0 \leq i \leq 15$ and $0 \leq j \leq 15$, where CENR* is fixed and Λ_{ij} is a random offset. At the beginning of each session, each $\Lambda_{i,j}$ is chosen uniformly at random from the interval $[-\lambda, \lambda]$ for $\lambda > 0$, and remains fixed throughout the session. The primary and secondary relays are also assigned at the beginning of each session according to the method of Appendix D. The performance of CFC-based broadcast and the conventional broadcast scheme are shown in Figure 4.13 for Rayleigh fading on the links and $\lambda = 10$ dB. We see from the figure that, on average, our network-coded broadcast schemes provide up to 12 dB of performance benefit over conventional broadcast. The four modes of network-coded broadcast performs within 1.5 dB of one another. The relatively small difference between the throughput of the network-coded broadcast strategies is a result of efficient relay selection.

Chapter 5

Two-Hop Relay Networks

In this chapter, we examine the use of network coding in a special type of multicast network, which we refer to as the *two-hop relay network*. It is a network in which a source node tries to transfer a file to a set of remotely-situated destinations with the help of one or more intermediate nodes that do not need the file themselves. The SNR of the direct link from the source to each of the destinations is too poor to permit communications. Therefore, for the networks considered in this chapter, the destinations are unable to overhear any of the transmissions made by the source. The only way to get the file to the destinations is for the source to deliver the file to the relays and then the relays deliver the file to the destinations.

An example of a two-hop relay network with two intermediate nodes is shown in Figure 5.1. In this example, node S is the source, nodes R_1 and R_2 are intermediate nodes, and nodes D_1 through D_N are destinations. The links from S to D_1 through D_N do not support packet delivery. All other links in the network support communications, although the links are not necessarily of the same quality. Suppose that the link from S to R_1 has higher SNR than the link from S to R_2. Also suppose that the link from R_1 to a destination has higher SNR than the link from node R_2 to the same destination. In this network, conven-

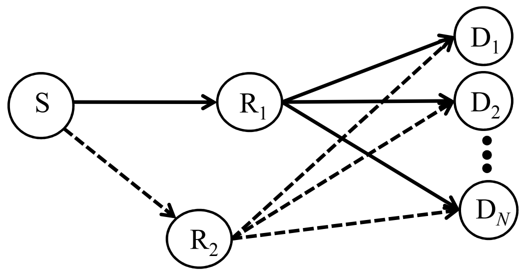


Figure 5.1: An example of a two-hop relay network.

tional approaches that utilize ARQ typically permit only node R_1 to forward the file from S to the destinations because R_1 has better incoming and outgoing links compared with R_2. In contrast, if one of modes TS, CS, and AS of network-coded file transfer described in Chapter 4 is applied to the two-hop relay network, both intermediate nodes can be utilized as relays to achieve higher throughput. For example, node R_1 and R_2 can be designated as primary and secondary relays, respectively, and then mode TS or CS can be used. Alternatively, mode AS can be employed, which allows any node in the network to function as a secondary relay node and transmit NC packets to its neighbors. Recall that each of these modes permits a relay node to send NC packets only after the node has decoded the file. This choice is motivated by the fact that, for half-duplex radios, file decoding at a node is delayed if the node transmits NC packets to its neighbors before decoding the file. This, however, is not a concern for relay nodes that do not need the file. Such nodes may be allowed to occasionally stop receiving packets from upstream nodes and forward NC packets to neighbors even when they have not obtained the complete file themselves. In the following section, we describe two modes of network-coded file transfers in which both intermediate nodes cooperate to send NC packets to the destinations without waiting to receive all packet in the file.

5.1 Modes of Operation

The description here is for a network with two intermediate nodes, similar to the one in Figure 5.1. While the example in Figure 5.1 depicts a topology in which each of the incoming and outgoing links for one intermediate node is better than the corresponding incoming and outgoing links for the other intermediate node, this is not a requirement for our methods. It is, however, assumed that each intermediate node has operational links to each of the destinations. General multicast networks in which more than two intermediate nodes may be required to reach all destinations are considered in Chapter 6.

In the description below, we assume that one intermediate node has been designated as the primary relay and the other as the secondary relay. As in our network-coded broadcast techniques, a number of strategies can be used to assign the primary and the secondary relay. One possible approach is to first define the cost of using a relay as the sum of the costs of that relay's incoming link from the source and the outgoing links to the destinations, and then use the relay with the lower cost as the primary and the other relay as the secondary.

Suppose that the file at the source consists of K information packets. Any network-coded system in which the relay nodes must receive, decode, and then encode such a file requires each relay node to have sufficient space in its buffer to store at least K packets. If a relay node is also a destination, then it is reasonable to assume that the node has enough memory to store the incoming NC packets. However, for intermediate nodes that are not destinations but merely serve as relays, it is conceivable that some nodes do not have enough buffer space to accommodate all K packets, especially when K is large. With this in mind, we suggest two modes of network-coded file distribution for two-hop relay networks. In the first mode, we assume that the intermediate nodes may not have enough memory to store the file. We refer to this mode as the low-memory (LM) mode. In the other mode, we assume that the intermediate nodes have sufficient memory to decode and store the file. This mode is referred to as the network-decoding-at-relays (DR) mode. In both modes, each relay alternates between receiving and transmitting, as described below.

5.1.1 Mode LM

Let W_p and W_s denote the number of NC packets the primary and the secondary relays are capable of storing in their respective buffers. Also, let W be an integer such that $1 \leq W \leq \min\{W_p, W_s, K\}$. The source begins the session by applying network coding to the information packets in the file and sending NC packets to the primary relay. The primary relay continues to receive until it recovers W packets from the source. Then it stops receiving and makes W transmissions to the destinations. The relay has two options: It can forward the NC packets to the destinations or it can apply network coding to the NC packets and transmit W random linear combinations of the NC packets. We will discuss these options in more detail later. After making W transmissions, the primary relay returns to its receive mode and stays there until it recovers another batch of W NC packets before switching to its transmit mode again.

When the primary relay enters its transmit mode, the source begins sending NC packets to the secondary relay, which continues to receive those packets until the primary relay returns to its receive mode. Then, as the primary relay receives from S, the secondary relay sends the recently-obtained NC packets to the destinations. The amount of time the secondary relay spends receiving or transmitting packets in each cycle is dictated by the primary relay. When the primary relay switches from transmit to its receive mode, it instructs the secondary relay by means of a control packet to switch to its transmit mode, and vice versa. As a result, unlike the primary relay which always accumulates W NC packets at the end of each reception cycle and then proceeds to make exactly W transmissions, the secondary relay may sometimes get more transmission opportunities than the number of NC packets in its buffer. In such situations, the secondary relay makes only as many transmissions as there are NC packets in the buffer and does not utilize the additional transmission opportunities.

The relays continue to take turns to transmit and receive batches of NC packets until each of the destination nodes is able to decode the file. The relays are not required to decode the file; therefore, each relay node can discard the most recent batch of NC packets from its buffer as soon as the batch has been transmitted, thereby avoiding potential buffer overflows. In fact, if the relays forward the NC packets rather than combine them before transmitting, then each NC packet can be discarded immediately after it has been forwarded.

5.1.2 Mode DR

Mode DR also requires the relay nodes to take turns transmitting and receiving batches of NC packets in the same fashion as mode LM. However, instead of discarding the NC packets after they have been transmitted, the primary relay stores them and decodes the file once enough packets have been accumulated. If the relay is able to decode the file before the destinations are able to do so, it instructs the source and the secondary relay to stop transmitting, applies network coding to the decoded information packets, and continues to send network-coded packets to the destinations until each destination decodes the file. For fountain-coded relaying, continued-fountain coding (CFC) is used to prevent the relay from producing fountain-coded packets that one or more destinations have already received. Before the primary relay starts the fountain encoder, it broadcasts a control packet to the destinations and each destination responds with a reply packet that contains the largest sequence number that the recipient has received thus far. The relay starts its fountain encoder from a state that is one higher than the maximum of the sequence numbers reported by the destinations, which ensures that no duplicate fountain-coded packets are sent to any of the recipients.

5.2 Choosing a Value for W

The choice of the parameter W is critical to the performance of the relaying schemes. For example, consider a hypothetical situation in which there are no packet erasures on any of the incoming or outgoing links for either intermediate node and suppose that all packets are of equal duration. The session completion time for this network is equal to the duration of $K+W+\varepsilon$ channel packets, where ε is the number of excess packets required to decode the network code at the destinations. Therefore, smaller values of W result in shorter sessions and higher throughput. Of course, the session will take longer than $K+W+\varepsilon$ channel-packet durations if there are packet erasures on the links. Also, not all channel packets are necessarily of the same duration when the adaptive transmission protocol is used. However, we have found that smaller values of W result in higher throughput even for practical networks and $W=1$ provides the best performance. But making W very small leads to an increase in the complexity of implementation. The smaller the value of W, the greater the number of times the relays must switch between transmitting and receiving, and the greater the amount of coordination between the relays that is required.

5.3 Forwarding vs. Network Coding at the Relays

When a relay node transmits NC packets to the destinations prior to decoding the file, it can either forward the NC packets stored in its buffer or apply network coding and send random linear combinations of the buffered NC packets. When fountain coding is employed, performing random linear combinations of fountain-coded packets at the relays can alter their degree distribution and thereby increase the complexity of decoding at the destinations. Therefore, only forwarding is used for fountain-coded packets. For RLNC, network coding at the relays does not affect the decoding complexity; therefore, a relay has the option to either forward the NC packets or send random linear combinations of the buffered NC packets. In our description, we use the abbreviations FR and NR to indicate *forwarding by the relays* and *network coding by the relays*, respectively.

5.4 Reporting of Decoding Completion for RLNC

In our general model for RLNC described in Chapter 2, each recipient reports the decoding of a generation to the sender by means of an acknowledgement packet. The sender stops sending NC packets from a generation that has been decoded by each of its next-hop neighbors. Because the destinations in a two-hop relay network receive NC packets from two relay nodes, and because each relay must alternate between receiving and transmitting packets, the destinations may be able to decode a generation before either of the relays is able to decode it. Consider a generation that has been decoded by all destinations but not by the relay nodes. Since the destinations do not provide any feedback to the source, the source continues to send NC packets from that generation, which reduces the throughput of the file transfer. To circumvent this situation, we slightly modify the reporting strategy for two-hop relay networks. In the modified method, when a relay node learns that each of the destinations has decoded a generation, it sends a proxy acknowledgement to the source. In response, the source stops sending NC packets from that generation.

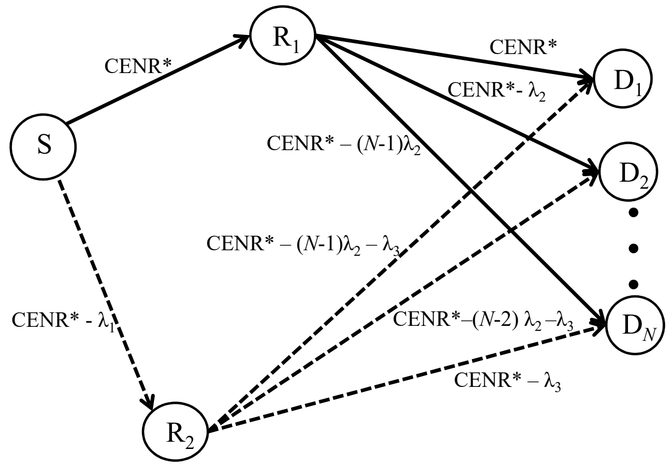


Figure 5.2: Simulation model for a two-hop relay network.[22]

5.5 Performance Results

We examine the distribution of a file of 500 information packets in the network of Figure 5.2. The network has N remote destinations D_1 through D_N. Nodes R_1 and R_2 are the primary and the secondary relay node, respectively, and they do not need the file themselves. The nominal values of CENR for the links from source S to relays R_1 and R_2 are CENR^* and $\mathrm{CENR}^* - \lambda_1$, respectively, whereas the nominal values of CENR for the incoming links to destination D_i from relays R_1 and R_2 are $\mathrm{CENR}^* - (i-1)\lambda_2$ and $\mathrm{CENR}^* - (N-i)\lambda_2 - \lambda_3$, respectively, for $\lambda_1, \lambda_2, \lambda_3 \geq 0$.

The session throughput of CFC-DR is shown in Figure 5.3 for four different values of the parameter W. The network has 10 destination nodes, and the parameters λ_1, λ_2, and λ_3 are 5 dB, 0.5 dB, and 0 dB, respectively. Each link in the network experiences Nakagami-m fading with $m = 2.5$. The figure also shows the performance of a conventional file-distribution strategy that employs ARQ for retransmissions and uses R_1 as the only relay node. A TPC of rate 0.472 with QPSK is used for all transmissions for both network-coded and conventional file transfers. We observe that for each value of W, CFC-

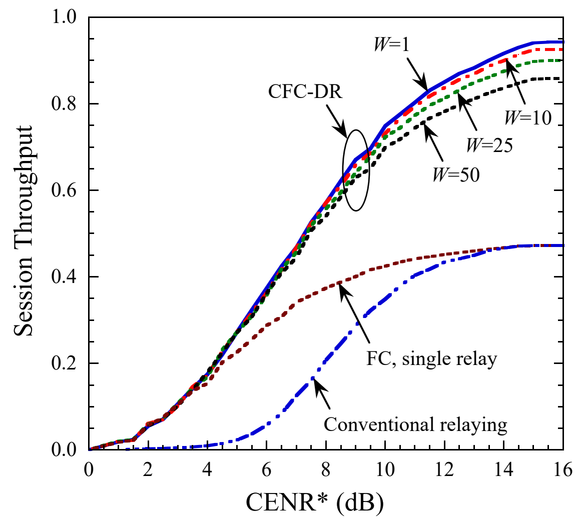


Figure 5.3: Session throughput of fountain-coded file distribution in the network of Figure 5.2. (A TPC of rate 0.472 with QPSK is used for transmissions; $N = 10$, $\lambda_1 = 5$ dB, $\lambda_2 = 0.5$ dB, $\lambda_3 = 0$ dB, and $m = 2.5$.)

DR significantly outperforms the conventional scheme. As expected, a smaller value of W results in higher throughput. But as discussed in Section 5.2, making W very small increases the implementation complexity. Therefore, we use $W = 25$ in all subsequent evaluations of the relaying schemes. This value of W provides a throughput that is within 5% of the throughput for $W = 1$.

Figure 5.3 also shows the throughput of a fountain-coding strategy that uses only one relay node. As in conventional relaying, this strategy routes all packets to the destinations by way of R_1; however, it avoids the retransmission of erased packets with the help of fountain coding. Note that this strategy is identical to mode NS from Chapter 4. Even though both FC-based relaying with a single relay and the conventional relaying scheme allow only R_1 to send packets, the former provides a performance benefit of about 3.5 dB over the latter across a large range of throughput values. This demonstrates the advantage that network coding offers over conventional ARQ-based relaying in a multicast setting. With ARQ, an erased channel packet must be retransmitted until all recipients recover it.

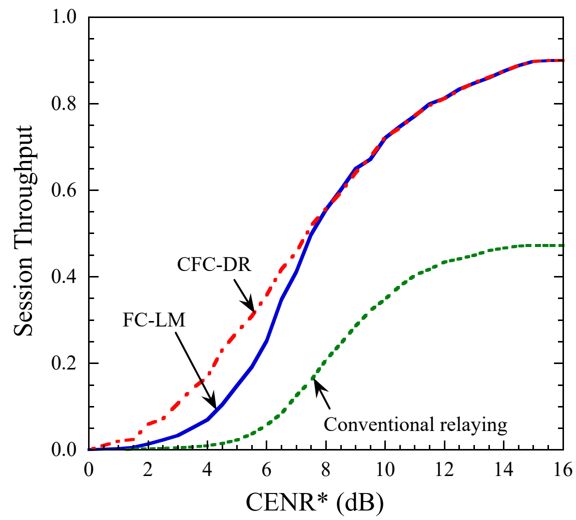


Figure 5.4: Comparison of CFC-DR and FC-LM in the network of Figure 5.2. (A TPC of rate 0.472 with QPSK is used for transmissions; $W = 25$, $N = 10$, $\lambda_1 = 5$ dB, $\lambda_2 = 0.5$ dB, $\lambda_3 = 0$ dB, and $m = 2.5$.)

The retransmissions carry no new information for the destinations that have already recovered the packet, and the throughput suffers as a consequence.

In Figure 5.4, we compare the performance of CFC-DR and FC-LM. We observe that both modes give the same performance at high SNR. When the links have high SNR and hence a low probability of packet erasure, it is unlikely that the primary relay node will receive enough packets to decode the file before the destinations are able to do so. Recall that if the primary relay does not get an opportunity to decode the file, the operation of CFC-DR is identical to that of FC-LM. However, at lower SNR, the primary relay in CFC-DR is sometimes able to decode the file, apply network coding to the information packets, and become the sole transmitter in the network. As we see from the figure, this leads to a substantial performance improvement in the low-SNR region. Although inferior to CFC-DR, FC-LM still significantly outperforms conventional relaying.

The performance of RLNC is shown in Figure 5.5. Recall that if RLNC is used instead of fountain coding, the relay nodes can either forward NC packets (FR mode) or

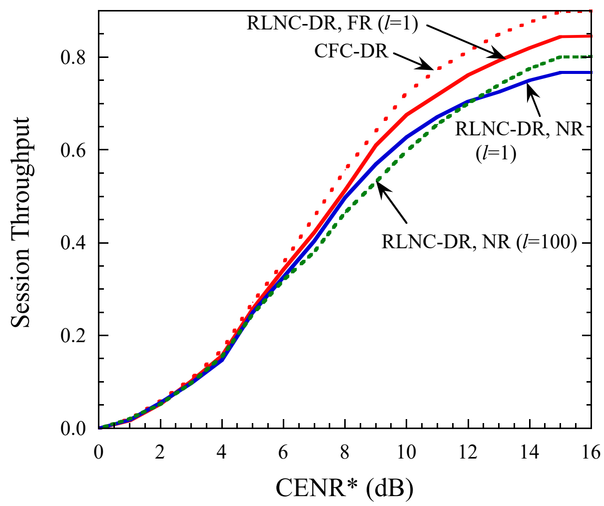


Figure 5.5: Performance of RLNC in the network of Figure 5.2. (A TPC of rate 0.472 with QPSK is used for transmissions; $N = 10$, $\lambda_1 = 5$ dB, $\lambda_2 = 0.5$ dB, $\lambda_3 = 0$ dB, $m = 2.5$.)

transmit random linear combinations of buffered NC packets (NR mode). We plot the performance of both modes in Figure 5.5. For comparison, the throughput of fountain-coded relaying is also shown. The relays operate in the DR mode for both CFC and RLNC. For RLNC, the file is divided into 5 generations of 100 packets each. The generation for each transmission is chosen according to the round-robin strategy described in Appendix E. This strategy employs an integer parameter $l > 0$. The sender chooses generations in a round-robin fashion, sending l consecutive NC packets from a generation before proceeding to the next. Once K transmissions have been made, l is set to 1. For the FR mode of RLNC, we have found that using $l = 1$ throughout the session gives the best throughput over the entire range of SNR. Therefore, we show results for only that value of l. The situation is less straightforward for mode NR. At high SNR when there are few or no packet erasures on the links, larger values of l perform better. Because the relay nodes transmit random linear combinations of buffered NC packets, having fewer packets from a generation in the buffer leads to greater likelihood of forming non-innovative packets.

When l is small, the relays have only a small number of packets from each generation stored in their buffers during the early phase of the session. As a result, they frequently send non-innovative NC packets to the destinations. If l is large, the relays receive packets from fewer generations during the initial phase but has more packets per generation. For example, when $l = 1$, $W = 25$, and the links have very high SNR, the primary relay obtains 5 NC packets from each generation before it gets its first opportunity to transmit. The relay then proceeds to make 25 transmissions, 5 from each generation. For each of these 5 transmissions, the network encoder at the relay must operate on only 5 NC packets. On the other hand, when $l = 100$, the primary relay obtains 25 NC packets from the first generation and none from the other generations before it gets its first opportunity to transmit. The relay then makes 25 transmissions, each of which is a random linear combination of 25 packets. Because the network encoder operates on a larger set of NC packets, the fraction of non-innovative packets generated is expected to be smaller in this case. This is true at lower values of SNR as well; however, the problem of correlated packet erasures explained in Appendix E dominates in that region. As a result, the performance of the NR mode of RLNC gives higher throughput for $l = 1$ than for $l = 100$ at low and moderate values of SNR. We have found that, regardless of the value of l, the NR mode's performance does not exceed that of the FR mode with $l = 1$. Therefore, we use mode FR with $l = 1$ for all subsequent evaluations of RLNC in this chapter.

In Figure 5.6, we compare the throughput of RLNC-DR and RLNC-LM. The two modes provide similar performance at high SNR but RLNC-DR outperforms RLNC-LM at low SNR, which is consistent with the observation made regarding CFC-DR and CFC-LM.

The performance of CFC-DR in the same network is shown in Figure 5.7 for two different values of the Nakagami parameter m, namely $m = 1$ and $m = 2.5$. For each value of m, we compare the throughput that is obtained when the senders use the AMCC protocol for transmissions with the throughput of two fixed-rate transmission schemes. We observe that the AMCC protocol provides much higher throughput than fixed-rate transmissions.

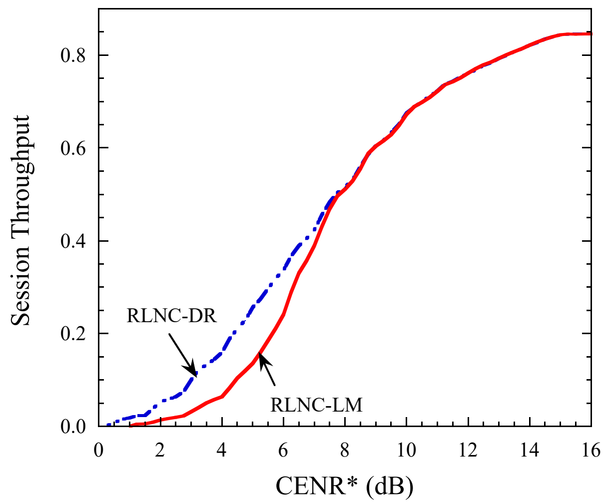
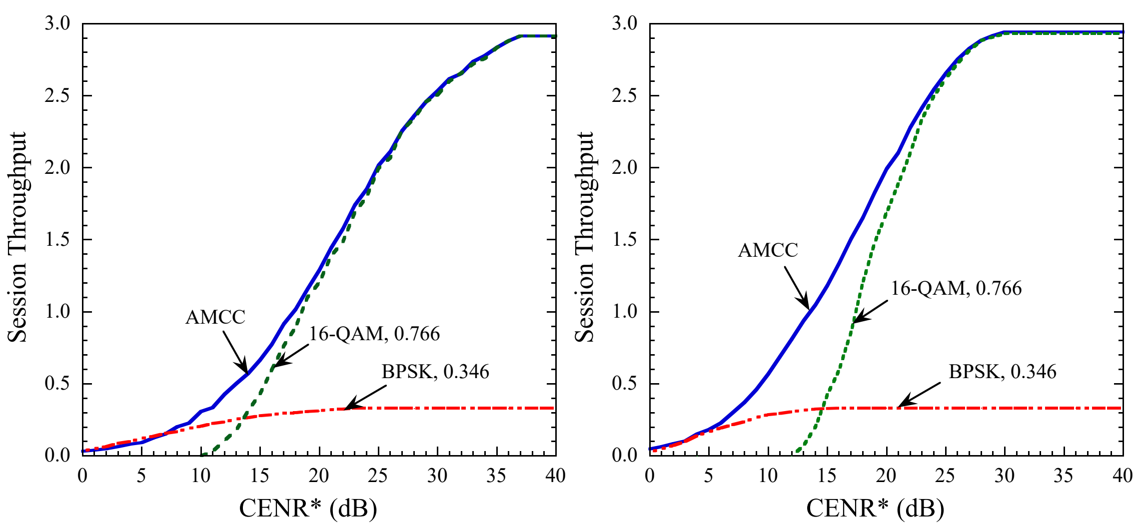


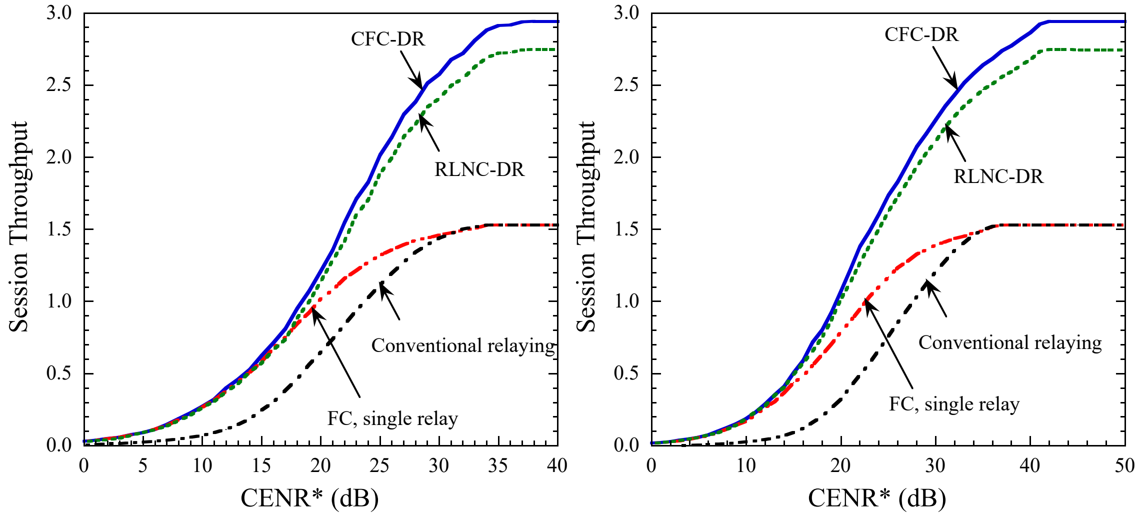
Figure 5.6: Comparison of RLNC-DR and RLNC-LM in the network of Figure 5.2. (A TPC of rate 0.472 with QPSK is used for transmissions; $W = 25$, $N = 10$, $\lambda_1 = 5$ dB, $\lambda_2 = 0.5$ dB, $\lambda_3 = 0$ dB, and $m = 2.5$.)



(a) $m = 1$.

(b) $m = 2.5$.

Figure 5.7: Comparison of AMCC and fixed-rate transmissions for fountain-coded file distribution in the network of Figure 5.2 ($N = 10$, $\lambda_1 = 8$ dB, $\lambda_2 = 0.75$ dB, $\lambda_3 = 3$ dB).



(a) $N=8$, $\lambda_1=10$ dB, $\lambda_2=1$ dB, $\lambda_3=0$ dB. (b) $N=20$, $\lambda_1=5$ dB, $\lambda_2=0.5$ dB, $\lambda_3=5$ dB.

Figure 5.8: Performance of different file-distribution techniques in the network of Figure 5.2 with Rayleigh fading on the links.

We can summarize the results so far as follows: CFC-DR outperforms CFC-LM, and RLNC-DR that employs the FR strategy and round-robin generation selection with $l=1$ outperforms other modes of RLNC. Also, network coding with AMCC gives higher throughput then network-coding without AMCC. We plot the session throughput of the best-performing modes of CFC and RLNC in Figure 5.8 for two networks. One of the networks has 8 destination nodes and the other has 20 destination nodes. The performance of conventional relaying and FC-based relaying with a single relay node are also shown. For both networks, CFC slightly outperforms RLNC while both CFC and RLNC significantly outperforms conventional relaying and FC-based relaying with a single relay node. A comparison of the two figures reveals that the performance gap between conventional relaying and network-coded relaying become larger as the number of destinations grows. The gap is as much as approximately 5 dB for 8 destination nodes, whereas it increases to about 7 dB for 20 destination nodes. The probability that a packet transmitted by a relay node is

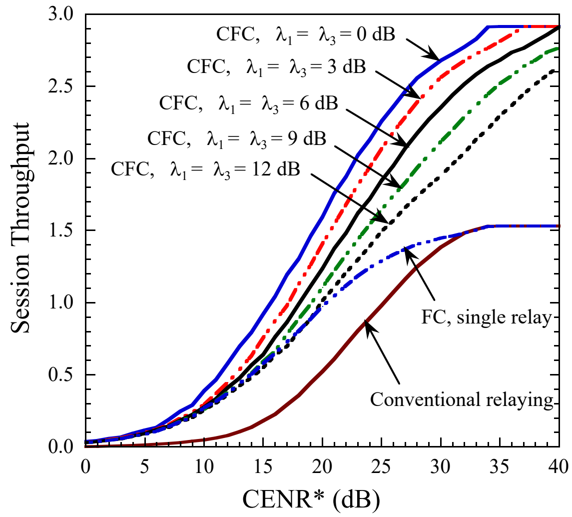


Figure 5.9: Performance of different file-distribution techniques in the network of Figure 5.2 with 15 destinations and Rayleigh fading on the links. The value of λ_2 is kept constant at 0.5 dB.

erased at one or more destination nodes increases as the number of destinations grows. For ARQ-based file transfers, this results in a larger number of retransmissions and reduces the throughput.

In Figure 5.9, we examine how the performance of CFC is affected when the nominal quality of the secondary relay's incoming and outgoing links are varied. We consider a two-hop relay network with 15 destination nodes. The value of λ_2 is kept fixed at 0.5 dB. We carry out performance evaluations for 5 different values of λ_1 and λ_3. An increase in λ_1 and λ_3 results in decreased ability of the destinations to obtain NC packets by way of the secondary relay node. At the extreme, when either λ_1 or λ_3 is large enough to render the incoming or outgoing link for the secondary relay incapable of delivering packets, the performance of the two-hop relaying scheme is expected to become approximately the same as that of the FC-based scheme with single relay. We observe from the figure that, as expected, $\lambda_1 = \lambda_3 = 0$ dB provides the best throughput and the throughput decreases as λ_1 and λ_3 are increased. The worst-case scenario in the figure corresponds to $\lambda_1 = \lambda_3 = 12$ dB.

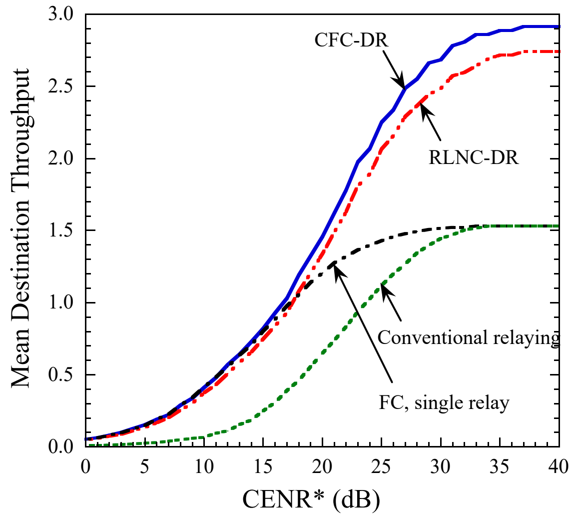


Figure 5.10: Mean destination throughput of different file-distribution techniques in the network of Figure 5.2 with Rayleigh fading on the links ($N = 8$, $\lambda_1 = 10$ dB, $\lambda_2 = 1$ dB, $\lambda_3 = 0$ dB).

But even for this fairly large offset, we notice that network-coded relaying with two relays provides significant performance improvements over the single-relay schemes.

For each of the two-hop relay networks we examined, we found the mean destination throughput to display the same trends as the session throughput. As a representative example, the mean destination throughput of different relaying schemes is shown in Figure 5.10 for the network with 8 destination nodes and offsets $\lambda_1 = 10$ dB, $\lambda_2 = 1$ dB, and $\lambda_3 = 0$ dB.

Chapter 6

General Multicast Networks

We now combine the insights gained from the previous two chapters to develop network-coded file-distribution strategies for general multicast networks. As for the broadcast schemes of Chapter 4, our approach permits recipients to overhear packets and allow nodes other than the primary relays to send network-coded packets after they have decoded the file. But if a node is not a destination, then it is also permitted to forward network-coded packets before it has received the complete file.

As before, we employ a spanning tree for the file transfer and we call each intermediate node in the tree a primary relay. Also, the nodes are assigned secondary relays whenever possible. Recall that the secondary relays provide alternate paths from the source to the nodes in the network. An example of a network with primary and secondary relays is shown in Figure 6.1. The nodes shown as circles are the primary relays whereas the hexagonal nodes are the secondary relays.

We introduce the following terminology to specify two types of relay operations and to describe the relationship between certain relay nodes:

- TAD (*transmit-after-decoding*) *relay*: A TAD relay does not transmit NC packet until it has decoded the file. When the relay decodes the file, it applies network coding to

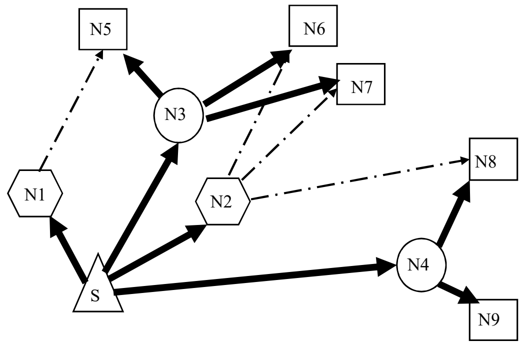


Figure 6.1: An example of primary and secondary relays.

the information packets and starts sending NC packets to its children.

- TBD (*transmit-before-decoding*) *relay*: A TBD relay is permitted to forward NC packets from its parent to its children before it has decoded the file. When operating in the TBD mode, the relay alternates between receiving and transmitting in order to forward batches of packets to its children.

- *Companion relay*: If two relay nodes have at least one child in common, then each relay is called a companion relay of the other. In Figure 6.1, N_1 and N_3 are companion relays of each other because they share the child N_5. The other two pairs of companion relays in the network are (N_2, N_3) and (N_2, N_4).

According to the terminology above, each relay node in the broadcast strategies of Chapter 4 is a TAD relay, whereas each relay node in the two-hop relay networks of Chapter 5 is a TBD relay.

We describe two modes of network-coded multicast. The modes differ in the way the primary relays send packets. In one of the modes, all primary relays act as TAD relays. We refer to this mode of operation as mode PTA. In the other mode, which we refer to as mode PTB, primary relays that are not destinations themselves may operate as TBD relays.

6.1 Mode PTA

In mode PTA, a spanning tree is first constructed using any suitable method and the secondary relays are assigned. Nodes that are not destinations are preferred choices for secondary relays. The source begins the session by applying network coding to the file and sending NC packets to its children. The primary relays operate in a manner identical to any of the broadcast modes of Chapter 4. Each primary relay acts as a TAD relay, i.e., it continues to receive NC packets until the file is decoded. Upon decoding the file, the primary relay broadcasts a control packet to its children announcing that it is ready to send NC packets. Each child responds to this announcement with a reply packet. The primary relay then applies network coding to the information packets and starts sending NC packets to its children. Secondary relays that are destinations themselves also act as TAD relays, which again is similar to how they operate in the broadcast modes of Chapter 4.

The difference between the broadcast strategies of Chapter 4 and mode PTA lies in the operation of the secondary relay nodes that are not destinations. Such nodes act as TBD relays, alternating between receiving and transmitting. There are two possible situations for such a secondary relay. First, suppose that the secondary relay is receiving packets from its own primary relay. In this case, the secondary relay receives from its parent until it recovers W_a NC packets and then forwards those packets to its children by making W_a transmissions. Then the relay returns to the receive mode to accumulate another W_a packets before it begins forwarding again. The secondary relay continues this process until the primary relays for all of its children decode the file and begin transmitting. The second situation arises when the secondary relay was assigned its own TBD secondary relay (referred to as the *secondary parent*) and is in the process of receiving packets from the secondary parent which itself is alternating between receiving and transmitting. In this case, the secondary relay coordinates its transmissions with its secondary parent. The relay stays in the receive mode when its secondary parent is forwarding NC packets and then forwards the recovered packets when its secondary parent is in the receive mode.

Whenever a node in the network decodes the file, it broadcasts a control packet offering to send NC packets to its neighbors. A neighbor that has not yet decoded the file accepts this offer and becomes a recipient if it is currently (a) not receiving packets from another sender or (b) receiving packets over an incoming link whose nominal SNR is lower than the nominal SNR of the incoming link from the new sender. Note that this behavior is identical to mode AS of network-coded broadcast described in Chapter 4. As in fountain-coded broadcast, the relay node uses CFC to prevent duplicate packets.

Now consider a recipient that has been assigned a TBD secondary relay. When the secondary relay alternates between receiving and transmitting, the best possible scenario for the recipient is that it recovers packets with an effective erasure probability of 0.5. This is because the secondary relay sends a batch of W_a packets and then returns to the receive mode for at least W_a packet durations before transmitting again. Suppose that the recipient has an operational link from an upstream primary relay node two or more hops away. Depending on the probability of packet erasure on this link, it may be beneficial for the recipient to overhear packets from that upstream node rather than to rely on the secondary relay. In mode PTA, the recipient attempts to overhear NC packets until the secondary relay starts forwarding packets. If the fraction of overhearing attempts that resulted in success exceeds a threshold Z_a, then the recipient decides to continue using the overhearing link and informs the secondary relay of this decision by means of a control packet. For a recipient that does not have a TBD secondary relay, the threshold test is not necessary. Such a recipient continues to overhear packets until a relay node decodes the file and starts sending NC packets.

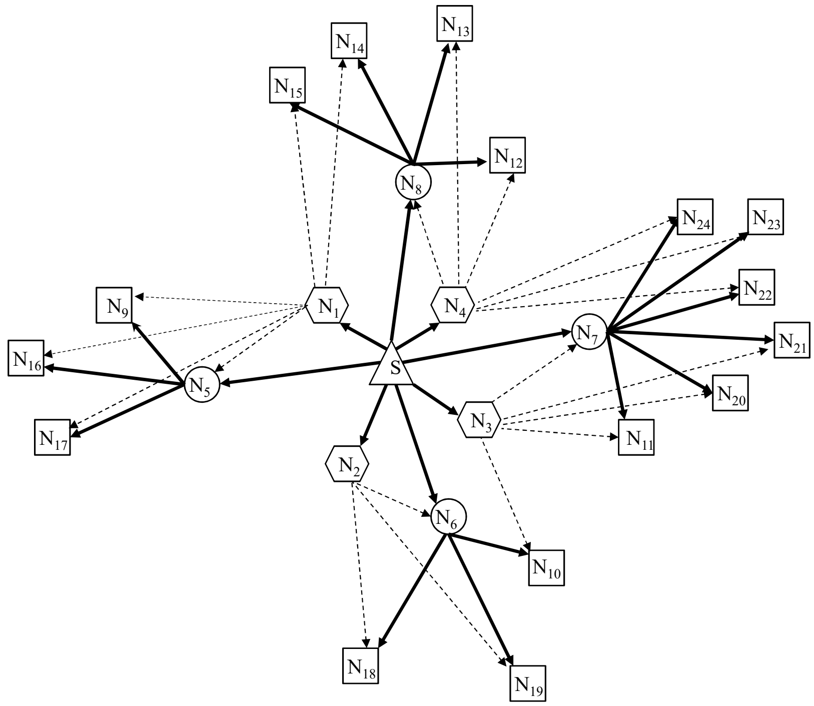


Figure 6.2: One possible relay assignment for mode PTA.

As an example of mode PTA's operation, consider the network of Figure 6.2. Node S is the source and the nodes shown as rectangles, i.e., N_9 through N_{24}, are the destinations. Nodes N_5 through N_8, which are depicted as circles, are the primary relays. Nodes N_1 through N_4, shown as hexagons, are the secondary relays. Node S sends NC packets from the file to N_1 through N_8, and the rectangular nodes attempt to overhear the transmissions. Because the secondary relays are not destinations, each acts as a TBD relay. Now consider the operation of the secondary relay N_1. When N_1 recovers W_a NC packets from S, it broadcasts a control packet informing nodes N_9 and N_{14}–N_{17} that it is ready to forward those packets. At this point, each of N_9 and N_{14}–N_{17} computes the ratio of the number of NC packets that it overheard from S to the total number of overhearing attempts it made. If that ratio is smaller than the predefined value of the threshold Z_a, the destination replies back to N_1 expressing its willingness to receive the forwarded NC packets. Otherwise, the destination continues overhearing NC packets from S. If at least one among N_9 and N_{14}–

N_{17} decides to receive packets from N_1, then N_1 begins to forward batches of W_a packets from S to the interested node(s). When the primary relay N_5 decodes the file, it applies network coding to the information packets and starts sending NC packets to N_9, N_{16}, and N_{17}, each of which then starts receiving from N_5, regardless of whether it was previously overhearing NC packets from S or receiving forwarded packets from N_1. Meanwhile, node N_1 continues to forward batches of W_a NC packets from S to N_{14} and N_{15} until N_8 decodes the file.

6.2 Mode PTB

Recall that in mode PTA, a recipient that is in the process of receiving NC packets from its TBD secondary relay must stay idle when the secondary relay returns to its receive mode to recover new NC packets from its parent. Consider a recipient for which neither the primary nor the secondary relay is a destination node. If we permit both relays to operate as TBD relays and allow them to take turns receiving and forwarding NC packets in a manner similar to the two-hop relay network of Chapter 5, the idle time at the recipient is greatly reduced; consequently, we might expect a significant reduction in the recipient's decoding-completion time. This is the central philosophy behind mode PTB, in which primary relays that are not destinations act as TBD relays and are paired with TBD secondary relays to form two-hop relay networks. As in mode PTA, the first step is to assign the primary relays by constructing a spanning tree. Next, the secondary relays are assigned, preferably from among the nodes that are not destinations, such that (a) a TBD secondary relay node has no more than one companion relay that is a TBD primary relay and (b) a TBD secondary relay and its TBD companion relay have the same parent. When relays are chosen in this manner, smaller two-hop relay networks are formed within the larger multicast network. An example is shown in Figure 6.3. The node placements in the network are identical

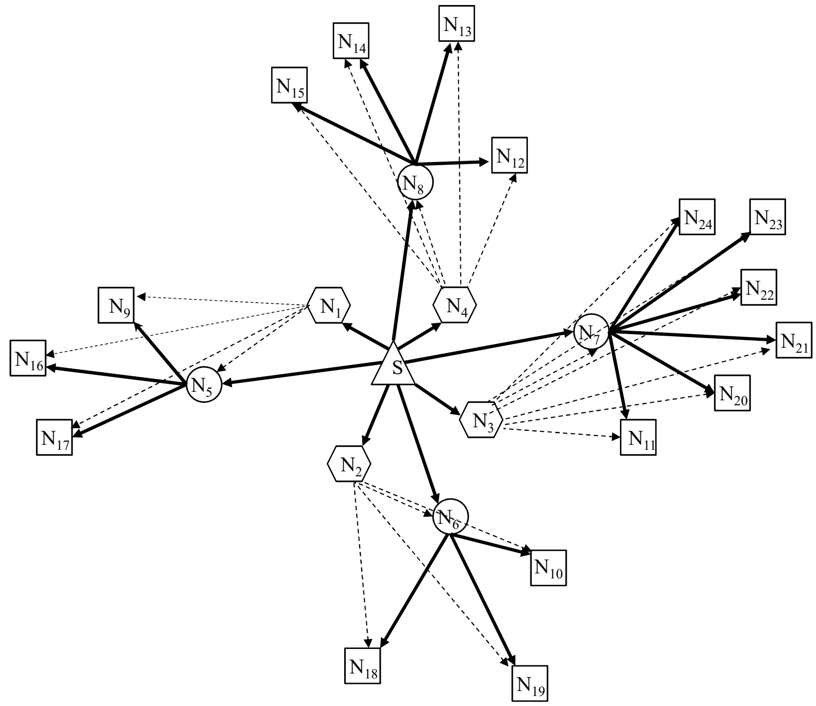


Figure 6.3: One possible relay assignment for mode PTB.

to the network of Figure 6.2, which was used to illustrate mode PTA. We make the same assumptions regarding destination and relay nodes, namely, only the rectangular nodes need the file and hence each relay node can operate as a TBD relay. The only difference is that the rectangular nodes are now assigned secondary relays in a way that ensures a secondary relay has no more than one companion relay. For example, node N_1 is no longer the secondary relay for nodes N_{14} and N_{15}, because that would require N_1 to have two companion TBD relays, namely N_5 and N_8, which violates condition (a) above.

The behavior of a relay node depends on whether it is a destination and on the nature of its companion relay. Each secondary relay that is not a destination acts as a TBD relay. Each primary relay that is not a destination and has at least one companion TBD secondary relay also acts as a TBD relay. A TBD primary relay and its TBD companion relay(s) take turns to receive and forward NC packets. The TBD primary relay stays in the receive mode until it recovers W_b NC packets from its parent and then forwards those packets to its children. Then it returns to the receive mode to accumulate another W_b packets, and so on. The TBD secondary relay receives packets from its parent when its TBD companion relay is in the transmit mode and forwards NC packets to the destinations when the companion relay is in the receive mode. A TBD secondary relay without a TBD companion relay operates as in mode PTA: It keeps forwarding batches of W_a NC packets to its children until the primary relays of the children begin transmitting. Primary relay nodes that are destinations act as TAD relays, i.e., they send NC packets only after they have decoded the file. Primary relays that are not destinations and do not have companion TBD relays also act as TAD relays.

As a result, depending on the network topology, a recipient may have either a TAD or a TBD primary relay. Also, the recipient may be assigned a TAD secondary relay, a TBD secondary relay, or no secondary relay at all. In addition, the recipient may also have an overhearing link from an upstream node two or more hops away and it may sometimes be more beneficial for the recipient to overhear packets until its primary relay decodes the file and starts transmitting. Depending on the type of relays it has been assigned, the recipient chooses between overhearing packets and receiving from the relay nodes as follows:

- TBD primary, TBD secondary: The recipient attempts to overhear NC packets until the primary relay starts forwarding. If at least a fraction Z_b of the overhearing attempts succeed, then the recipient continues to overhear; otherwise, it stops overhearing and utilizes the secondary relay. The value of Z_b is usually close to 1, which ensures that overhearing is preferred over the use of relay nodes only when the overhearing link is very strong.

- TAD primary, TBD secondary: The recipient attempts to overhear NC packets until the secondary relay starts forwarding. If at least a fraction Z_a of the overhearing attempts are successful, then the recipient continues to overhear; otherwise, it stops overhearing and utilizes the secondary relay. The value of Z_a is usually much lower than Z_b.

- TAD primary, TAD secondary or no secondary assigned: The recipient overhears packets until the primary relay decodes the file.

- TBD primary, TAD secondary or no secondary assigned: The recipient attempts to overhear NC packets until the primary relay starts transmitting. If at least a fraction Z_a of the overhearing attempts are successful, then the recipient continues to overhear; otherwise, it stops overhearing and utilizes the primary relay.

Whenever a node in the network decodes the file, it broadcasts a control packet offering to send NC packets to its neighbors. All of its neighbors that are currently not receiving packets or are receiving packets over poorer incoming links become the node's recipients. The node applies network coding to the file, using CFC if fountain coding is employed, and starts sending NC packets to the recipients.

Consider the application of mode PTB to the network of Figure 6.3. At the beginning of the session, the source S sends NC packets to the primary relays N_5 through N_8. The secondary relays do not attempt to receive these initial transmissions whereas the destination nodes try to overhear the packets. We focus on the relay pair (N_1, N_5) and their children N_9, N_{16}, and N_{17} as a representative example. When the session begins, N_1 stays idle while N_5 receives NC packets form S until it recovers W_b of them. In the meantime, N_9, N_{16}, and N_{17} try to overhear packets from S. Suppose that, for each of them, less than a fraction Z_b of the overhearing attempts are successful; hence, the nodes decide to stop overhearing. Therefore, N_5 forwards the W_b NC packets in its buffer to N_9, N_{16}, and N_{17}. While N_5 transmits, N_1 receives NC packets from S. Next, N_5 returns to the receive mode while N_1 forwards the packets in its buffer to N_9, N_{16}, and N_{17}. This process continues until either all three destinations decode the file or one of the relay nodes decodes the file, whichever occurs first.

Now consider another scenario in which N_5 is a destination but N_1 is not. In this case, N_5 operates as a TAD relay and does not send NC packets until it decodes the file. Because N_1 is not a destination, it acts as a TBD relay, forwarding batches of W_a NC packets to N_9, N_{16}, and N_{17} until N_5 decodes the file. As before, nodes N_9, N_{16}, and N_{17} try to overhear packets from S at the beginning of the session. When N_1 recovers the first set of W_a NC packets, each of N_9, N_{16}, and N_{17} computes the ratio of the number of NC packets it was able to overhear from S and the total number of overhearing attempts it made. If that ratio is smaller than the threshold Z_a, the node replies back to N_1 expressing its willingness to receive the forwarded NC packets. Otherwise, it decides to continue overhearing NC packets from S until an intermediate source becomes available.

The same threshold test is applied if N_1 is a TAD relay and N_5 is a TBD relay. Finally, if both N_1 and N_5 are TAD relays, then N_9, N_{16}, and N_{17} overhear packets from S until one of the relay nodes decodes the file and begins transmitting.

6.2.1 Mode PTB with Perfect Channel Knowledge (PTB-PCK)

In modes PTA and PTB, it is not necessary for the nodes to know the exact values of the SNR on their respective incoming links or the model of the fading. Now consider a hypothetical situation in which a node in mode PTB knows the nominal CENR of the overhearing link as well as the maximum attenuation that may be caused by fading on the link. This knowledge can be utilized to decide whether the node should receive the NC packets forwarded by the relays or try to overhear packets. In a hypothetical strategy referred to as mode PTB with perfect channel knowledge (PTB-PCK), a destination node with a TBD primary and a TBD secondary relay uses the overhearing link if and only if it knows that the probability of packet erasure when the link is in its deepest fade does not exceed a certain threshold P_T. For example, suppose that $P_T = 10^{-4}$ and that the sender uses the AMCC protocol with the code-modulation combinations of Table 1 in Appendix C. The combination with the highest information rate and the greatest probability of packet erasure at any given SNR requires a CENR of 11.98 dB to achieve a packet-erasure probability of 10^{-4}. Now suppose that the overhearing link experiences Rayleigh fading for which the maximum attenuation is 16 dB. The destination chooses to overhear packets if the nominal CENR of the link, denoted by C_{ov}, is such that $C_{\text{ov}} - 16\,\text{dB} \geq 11.98\,\text{dB}$, i.e., if $C_{\text{ov}} \geq 27.98\,\text{dB}$. Otherwise, the destination decides to receive NC packets forwarded by the relay nodes.

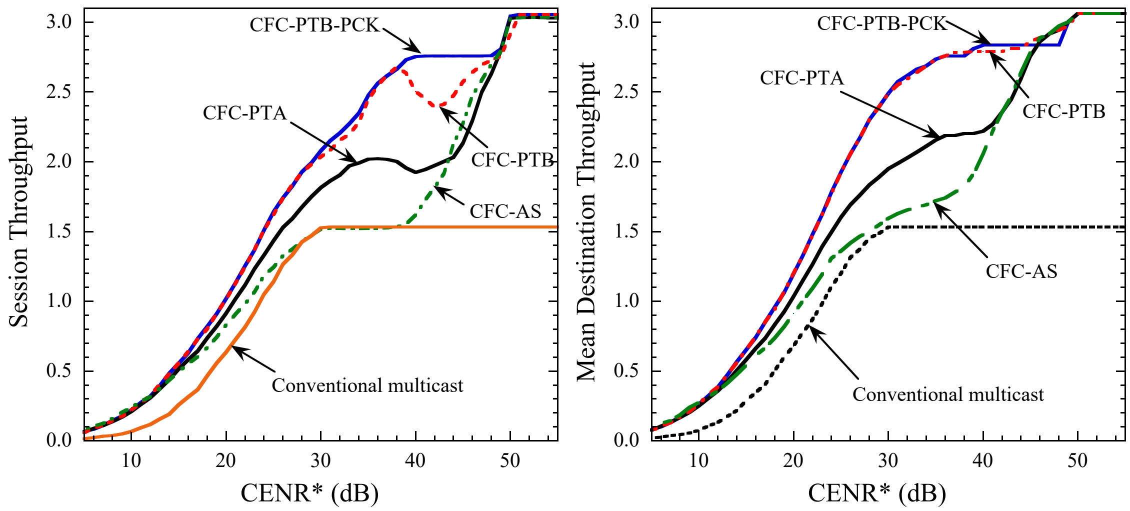
The parameter P_T is usually set to a small value. Mode PTB-PCK assumes that the quality of the overhearing link to a destination is unlikely to be better than that of the incoming links from the relay nodes. Therefore, unless the overhearing link is very strong, PTB-PCK decides to receive NC packets from the relays. A more effective decision than one simply based on comparing the nominal CENR to a threshold may be made by computing the expected throughput when overhearing is used and the expected throughput when the two relay nodes are used, and then utilizing the strategy that gives the higher expected throughput. However, such computations are quite complicated for most practical channel models and transmission schemes. (E.g., as shown in Chapter 7, they are not very simple even for fixed-rate communications over links that can be modeled by Markov chains with only two states.)

As mentioned above, the knowledge of the nominal CENR and the fade levels is utilized only by those nodes for which both primary and secondary relays operate as TBD relays. For a node that is receiving NC packets forwarded by its TBD secondary relay while the TAD primary relay waits to decode the file, the effective rate of data transfer from the secondary relay to the destination can be rather low even when the link from the secondary relay to the destination is very strong, because the secondary relay must periodically return to the receive mode, leaving the destination idle. In this situation, the choice between overhearing and using relay nodes cannot be made based simply on the nominal CENR of the overhearing link and its fade levels. Therefore, unless a destination in mode PTB-PCK has a TBD primary and a TBD secondary relay, it applies the threshold test described in Section 6.1 to the fraction of overheard packets. This is also the reason why we do not consider a PCK counterpart to mode PTA.

6.3 Performance Results

For our numerical results, the file at the source consists of 500 information packets. Unless otherwise specified, the parameters W_a, W_b, Z_a, Z_b, and P_T are set to 125, 50, 0.5, 1, and 10^{-4}, respectively. The AMCC protocol is used for all transmissions.

We first consider the network of Figures 6.2 and 6.3. For mode PTA, we assign the relays as shown in Figure 6.2, whereas mode PTB uses the relay assignment of Figure 6.3. Nodes N_9 through N_{24} are the destinations and none of the relay nodes need the file. We employ two offset parameters λ_1 and λ_2 to assign a range of nominal signal-to-noise ratios to the links in the network. The nominal CENR of the links from S to N_1 through N_4 is CENR^*, to N_5 through N_8 is $\text{CENR}^* - \lambda_1$, to N_9 through N_{12} is $\text{CENR}^* - \lambda_1 - \lambda_2/2$, and to N_{13} through N_{24} is $\text{CENR}^* - \lambda_1 - \lambda_2$, where $\lambda_1, \lambda_2 > 0$. The nominal CENR of the links to N_9 through N_{12} from their respective primary relays is CENR^* and the nominal CENR of the links to N_{13} through N_{24} from their respective primary relays is $\text{CENR}^* - \lambda_1$. In the relay assignment of Figure 6.2, the nominal CENR of the links to N_9 through N_{12} from their respective secondary relays is $\text{CENR}^* - 1.5\lambda_1$ and the nominal CENR of the links to N_{13} through N_{24} from their respective secondary relays is $\text{CENR}^* - 1.8\lambda_1$. Some of the destinations in mode PTB have poorer-quality links from their respective secondary relays



(a) Session throughput.

(b) Mean destination throughput.

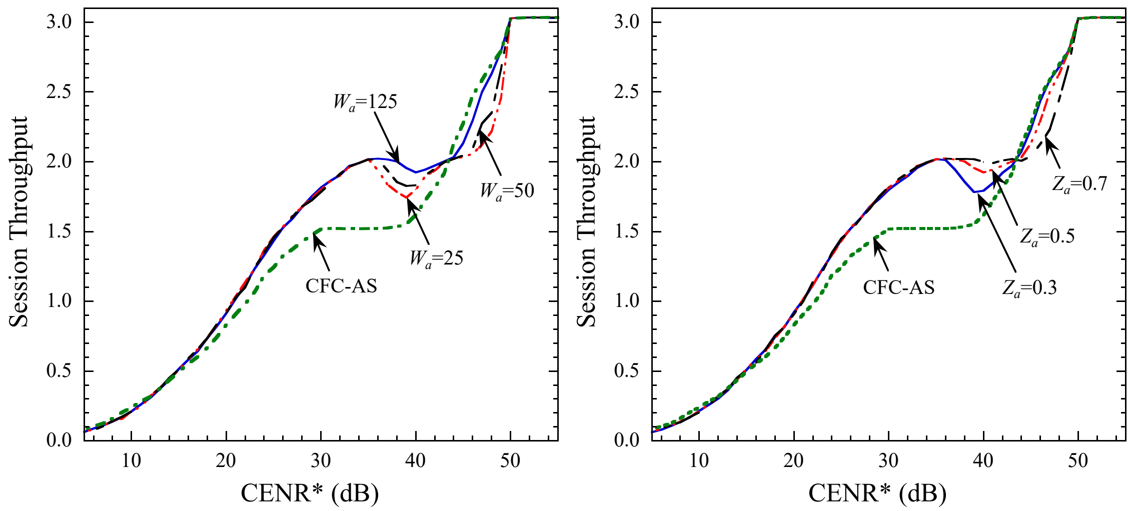
Figure 6.4: Throughput of different multicast schemes in the network of Figures 6.2 and 6.3 when none of the relay nodes need the file ($\lambda_1 = 10$ dB, $\lambda_2 = 20$ dB, $m = 2.5$).

than in mode PTA because additional constraints were imposed on the assignment of relay nodes for PTB.

Figure 6.4a shows the session throughput of PTA, PTB, and PTB-PCK modes of CFC when $\lambda_1 = 10$ dB and $\lambda_2 = 20$ dB and the Nakagami parameter m is 2.5. Their performance is compared with that of CFC-AS, which was found to be the best performer among the four modes in Chapter 4, and that of a conventional multicast strategy that uses ARQ for retransmissions. Modes PTA, PTB, and PTB-PCK each provides higher throughput than mode AS and conventional multicast. Mode PTA outperforms mode AS for all values of CENR* between 16 dB and 43 dB. Between 43 dB and 49 dB, there is a slight disadvantage to using mode PTA over mode AS; however, the reduction in throughput in this SNR range is quite small compared with the performance gains of PTA at lower SNR. The throughput gains for modes PTB and PTB-PCK over mode AS and conventional multicast are much greater than those for mode PTA. We observe that there is a dip in the throughput of mode PTB, and to a lesser extent, in that of mode PTA, when CENR* is in the range of 38 dB to 45 dB. Recall that the destinations in both these modes rely on the first few overheard packets to assess the quality of the respective overhearing links from the source and use a threshold test to decide between utilizing the relay nodes and receiving packets directly from the source. The dips in the throughput occur for CENR* values at which the destinations are more likely to make the wrong decision. This issue is discussed in more detail later. Because mode PTB-PCK does not rely on such assessments of the overhearing link, no dips are observed in its session throughput.

The mean destination throughput for the same strategies in the same network are shown in Figure 6.4b. The primary difference between these curves and those for the session throughput is that the curves for the mean destination throughput of modes PTA and PTB do not feature any dips. This shows that the occasional inaccurate assessments of the overhearing link has a less severe impact on the average decoding completion time at the nodes than it has on the session completion time.

In Figure 6.5a, we plot the performance of CFC-PTA for three values of W_a, namely 25, 50, and 125. The parameter Z_a is set to 0.5. For comparison, the session throughput of CFC-AS is also shown. The figure shows that a larger value of W_a results in higher throughput for CFC-PTA. This is a consequence of the increased accuracy of the threshold test employed by the destinations to choose between overhearing packets from the source and utilizing the secondary relay node. Recall that each destination in mode PTA overhears NC packets from the source until the secondary relay begins forwarding the first batch of NC packets. The fraction of overhearing attempts that succeeded is then compared with a threshold Z_a to decide whether the destination should continue to overhear or utilize the secondary relay. The larger the value of W_a, the more overhearing attempts a destination can make before the secondary relay begins transmitting. Consequently, the fraction of overhearing attempts that were successful is a more reliable indicator of the quality of the overhearing link when W_a is large.



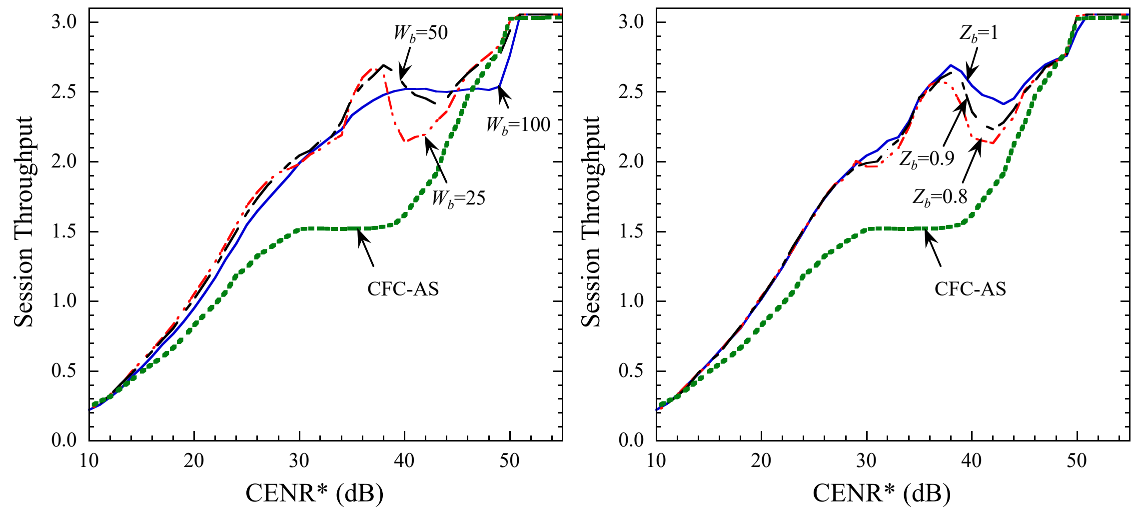
(a) $Z_a = 0.5$.

(b) $W_a = 125$.

Figure 6.5: Throughput of CFC-PTA in the network of Figure 6.2 for different values of W_a and Z_a ($\lambda_1 = 10$ dB, $\lambda_2 = 20$ dB, $m = 2.5$).

Notice that all four curves in the figure meet when CENR* is approximately 44 dB. This implies that, at this value of CENR*, the same throughput is achieved regardless of whether a destination chooses to receive packets forwarded by its secondary relay or continues to overhear packets from the source. When CENR* is below 44 dB, it is more beneficial for the destinations to receive packets from their secondary relays than to overhear, whereas above 44 dB, overhearing is preferable.

Figure 6.5b shows the performance of CFC-PTA for three values of the threshold Z_a when W_a is fixed at 125. It can be seen that larger values of Z_a provide higher throughput when CENR* is below 44 dB. As mentioned above, for CENR$^* < 44$ dB, using secondary relays is preferable to overhearing. The larger the value of the threshold Z_a, the more likely a destination is to favor the use of the secondary relay, and hence the higher the throughput. For CENR$^* > 44$ dB, on the other hand, it is more beneficial to rely on overhearing. Because making Z_a smaller increases the likelihood of destination nodes choosing to overhear packets instead of using their secondary relays, we observe an increase in the high-SNR



(a) Session throughput vs. W_b ($Z_b = 1$). (b) Session throughput vs. Z_b ($W_b = 50$).

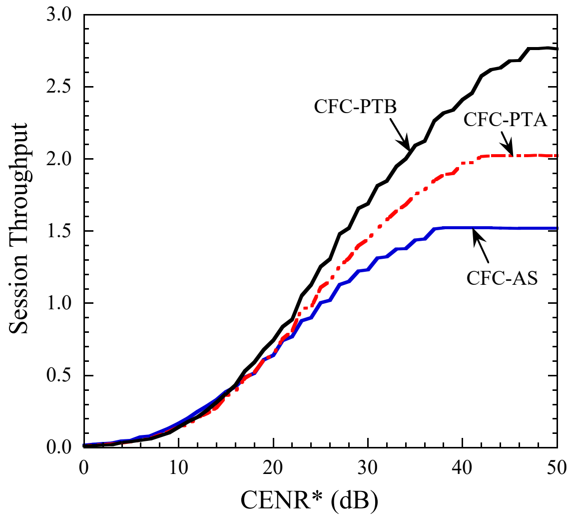
Figure 6.6: Throughput of CFC-PTB in the network of Figure 6.3 for different values of W_a and Z_a ($\lambda_1 = 10$ dB, $\lambda_2 = 20$ dB, $m = 2.5$).

throughput of CFC-PTA as the value of Z_a is reduced.

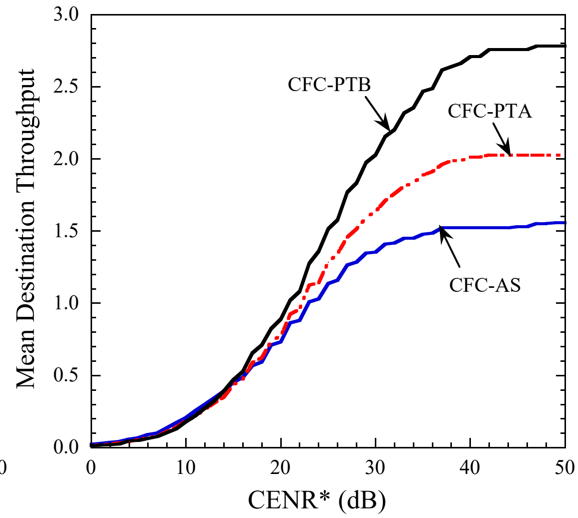
The session throughput of CFC-PTB for three values of W_b are shown in Figure 6.6a. The threshold Z_b is set to 1 for these results. As explained earlier for mode PTA, smaller values of W_b make a destination more prone to making the wrong choice between overhearing and receiving from relays. This produces the sharp dip in the throughput for $W_b = 25$ when CENR* is about 38 dB. On the other hand, making W_b too large also reduces the throughput due to the reasons explained in Section 5.2 of Chapter 5. Therefore, moderate values of W_b are more suitable for mode PTB.

Figure 6.6b shows the performance of CFC-PTA for three values of the parameter Z_b when W_b is fixed at 50. We see that $Z_b = 1$ gives the best session throughput. This is consistent with our observation that a destination with a TBD primary and a TBD secondary relay should choose to overhear packets only if the overhearing link is very strong.

In Figure 6.7, we plot the performance of the file-distribution strategies for the networks of Figures 6.2 and 6.3 with $\lambda_1 = 10$ dB, $\lambda_2 = 60$ dB, and Rayleigh fading on the



(a) Session throughput.

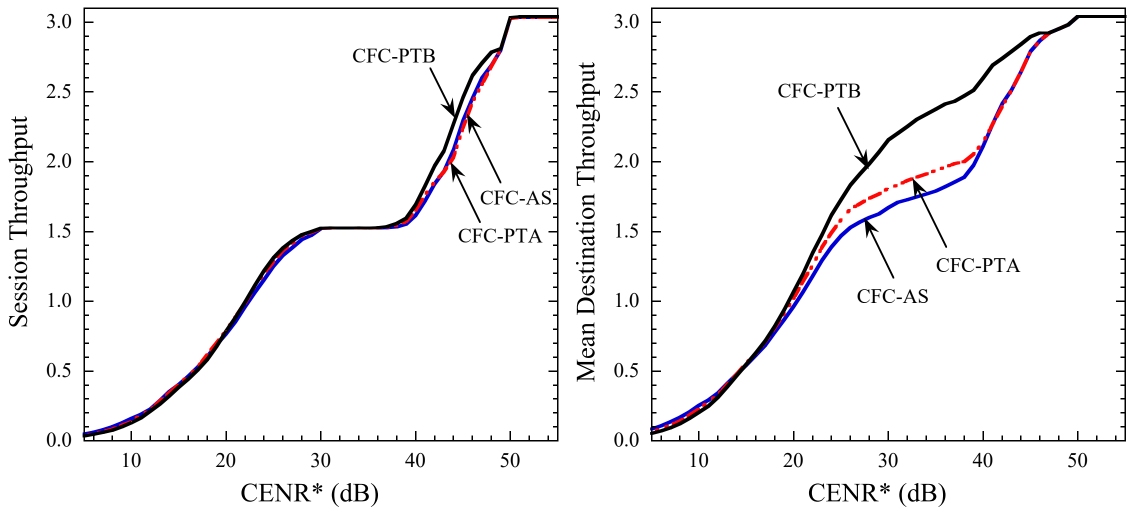


(b) Mean destination throughput.

Figure 6.7: Throughput of different multicast schemes in the network of Figures 6.2 and 6.3 when none of the relay nodes need the file ($\lambda_1 = 10$ dB, $\lambda_2 = 60$ dB, $m = 1$).

links. The large value of λ_2 renders the links from S to the destination nodes incapable of supporting any packet delivery; consequently, the destinations can no longer benefit from overheard packets. Modes PTA and PTB outperform mode AS for this scenario as well, with mode PTB providing significantly higher throughput than each of the other modes.

The scenarios that have been considered thus far are particularly suitable for modes PTA and PTB because none of the relay nodes need the file and for each remote node, there is at least one neighbor available to serve as a TBD secondary relay. Now we examine a less favorable situation by assuming that nodes N_1 and N_2 are destinations. Thus, for both modes PTA and PTB, some of the remote nodes are assigned secondary relays that do not act as TBD relays. The performance of the different modes are shown in Figure 6.8. It can be seen that the inability of the two secondary relays to forward NC packets before decoding the file has the effect of reducing the session throughput of modes PTA and PTB to almost the same level as that of mode CS. Recall that the session throughput depends on the decoding-completion time for the destination that is the last to obtain the file. For



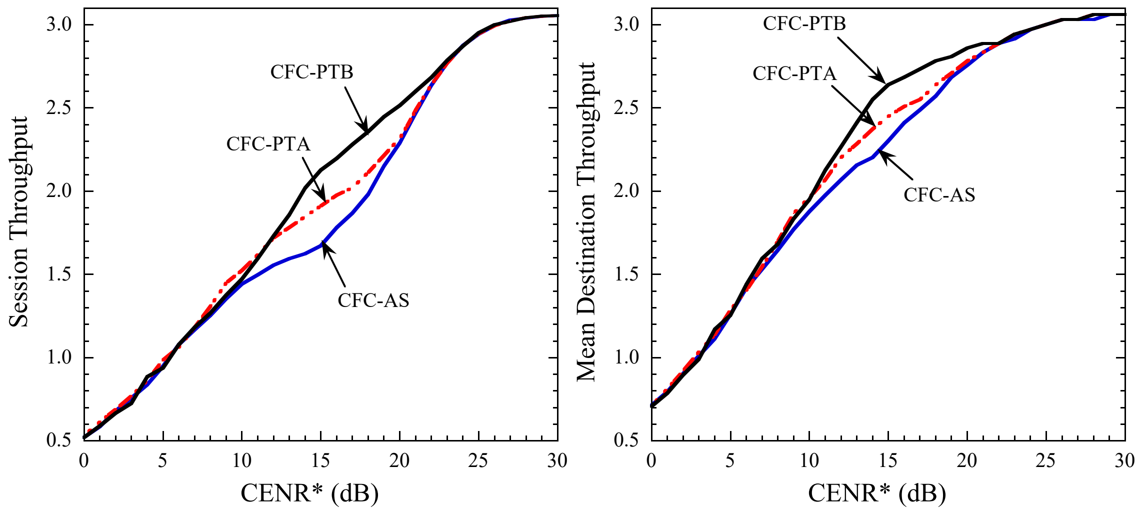
(a) Session throughput.

(b) Mean destination throughput.

Figure 6.8: Throughput of different multicast schemes in the network of Figures 6.2 and 6.3 when relay nodes N_1 and N_2 also need the file ($\lambda_1 = 10$ dB, $\lambda_2 = 20$ dB, $m = 2.5$).

modes PTA and PTB, it is very likely that one of the remote nodes that have N_1 or N_2 as their secondary relay will be the last node to decode the file. Because their respective relay nodes do not send NC packets before decoding the file, these remote nodes must rely on overhearing alone until a relay node decodes the file. Consequently, the average session completion times for modes PTA, PTB, and AS are all approximately the same. However, the decoding completion times for the remote nodes that have N_3 and N_4 as their secondary relays are much smaller for modes PTA or PTB than for mode AS. This is reflected in the mean destination throughput, which is higher for modes PTA and PTB than for mode AS. As before, mode PTB provides significantly higher mean destination throughput than the other modes.

Next, we examine the throughput of the multicast strategies averaged over random nominal link-quality assignments. Suppose there are 20 nodes in the network in addition to the source. The nodes are numbered N_0 through N_{20}. The nominal CENR of the link between node N_i and node N_j, where $0 \leq i \leq 20$ and $0 \leq j \leq 20$, is given by $\text{CENR}^* + \Lambda_{i,j}$,



(a) Session throughput.

(b) Mean destination throughput.

Figure 6.9: Throughput of different multicast schemes averaged over randomly generated network topologies with 20 nodes, out of which 5 are destinations ($\lambda = 10$ dB, $m = 2.5$).

where CENR* is fixed and $\Lambda_{i,j}$ is a random offset. At the beginning of each session, $\Lambda_{i,j}$ is picked uniformly at random from the interval $[-\lambda, \lambda]$ for some $\lambda > 0$. Then 5 of the nodes are chosen uniformly at random to be destination nodes. The primary and secondary relays are also assigned at the beginning of each session according to the relay-selection algorithm described in Appendix D. The offsets $\Lambda_{i,j}$ remain fixed throughout the session. The session throughput and the mean destination throughput of CFC-PTA, CFC-PTB, and CFC-AS are shown in Figure 6.9 for $\lambda = 10$ dB. We observe that, when CENR* is between 10 dB and 25 dB, CFC-PTB outperforms CFC-PTA, which in turn provides higher throughput than CFC-AS.

Chapter 7

Analytical Upper Bounds for Four-Node Networks

In this chapter, we outline an approach to deriving analytical upper bounds on the session throughput of file distribution in half-duplex packet radio networks. We illustrate our approach for three networks, each comprising four nodes. The first is a broadcast network in which one node wants to transfer a file to each of the three other nodes. In the second network, a node tries to deliver a file to two of the other nodes. The third network is a two-hop relay network with one destination node. We evaluate the bounds for links modeled by two-state Markov chains. The bounds are then compared with the simulated throughput of CFC-based file distribution in the same networks.

To simplify the analysis, we restrict attention to systems in which the same channel code and modulation format are used for all transmissions. For such systems, all channel packets are of equal duration and the expression for the session throughput given by (3.2) in Chapter 3 is equivalent to

$$\bar{S} = \frac{\rho K}{\overline{N}_s}, \tag{7.1}$$

where ρ is the information rate of the code-modulation combination in bits per modulation chip, K is the number of packets in the file, N_s is the session completion time expressed in terms of the number of channel-packet durations, and $\overline{N}_s$ denotes the expected value of N_s. Unlike in the rest of the dissertation, here we use the duration of one channel packet as the time unit, rather than the duration of one modulation chip. In (7.1), we have assumed that each destination in the network has at least one path from the source that is capable of transferring packets (such a path may be either a direct link from the source to the destination or a concatenation of links) and that a session ends only when all destinations obtain the file. If at least one destination has no path from the source, the session fails and the session throughput is trivially zero.

We utilize the concept of *capacity-achieving codes* (CAC) [25] in our analysis. Let $\mathcal{C}$ be the modulation-constrained AWGN channel capacity for the modulation format used for packet transmissions. A capacity-achieving code of rate r for the system is defined to be a binary channel code whose block error probability is $P_e = 0$ if $r < \mathcal{C}$ and $P_e = 1$ otherwise. The capacity $\mathcal{C}$ is a function of the SNR; therefore, the inequality $r < \mathcal{C}$ is equivalent to an inequality between CENR and the *capacity limit* Γ, where Γ is the greatest lower bound on the values of CENR for which $r < \mathcal{C}$.

Our approach is as follows: Given a strategy for file distribution, we obtain the probability mass function for the session completion time N_S for a hypothetical session in which (a) a recipient obtains a copy of the complete file as soon as it recovers a total of K channel packets and (b) a capacity achieving code is used for each transmission. We use the probability mass function for N_S to obtain $\overline{N}_S$, which gives a lower bound on the average session completion time for the given strategy. We then use the value of $\overline{N}_S$ in (7.1) to obtain an upper bound on the session throughput.

Note that criterion (a) does not specify whether a channel packet conveys one information packet, as in ARQ, or a combination of information packets, as in network coding.

In fact, neither ARQ nor network coding is guaranteed to meet this criterion. For ARQ, it is not possible to meet criterion (a) when a sender transmits packets to multiple recipients and different packets are erased at different recipients. For network coding, criterion (a) is violated whenever a sender produces a non-innovative packet. By employing criterion (a), we ensure that the resulting bounds provide an upper limit on the throughput of *any* file-delivery mechanism, including network-coding and ARQ.

In Section 7.1, we derive expressions for the probability mass function for the session completion times of three file-distribution strategies. The expressions are evaluated and the resulting bounds are compared with the throughput of network-coded file transfer in Section 7.2.

7.1 Analysis of the Session Completion Time

In each of the four-node networks, the nodes are numbered N_0 through N_3. In the analysis below, $T_{i,j}(k)$ denotes the number of packets node N_i must transmit to node N_j until the latter is able to recover k packets. Because packet erasures on a link are random, $T_{i,j}(k)$ is a random variable whose probability distribution depends on the distribution of the packet erasures on the link. We use another random variable $R_{i,j}(n)$ to denote the number of packets recovered by N_j when N_i makes n transmissions.

7.1.1 Broadcast File Distribution

Consider the network of Figure 7.1 and suppose that node N_0 wants to transfer a file of K information packets to nodes N_1, N_2, and N_3. The expression next to each link in the figure denotes the nominal CENR of the link. Our objective is to find an upper bound on the session throughput of a class of broadcast file-distribution techniques in which nodes behave in the following manner:

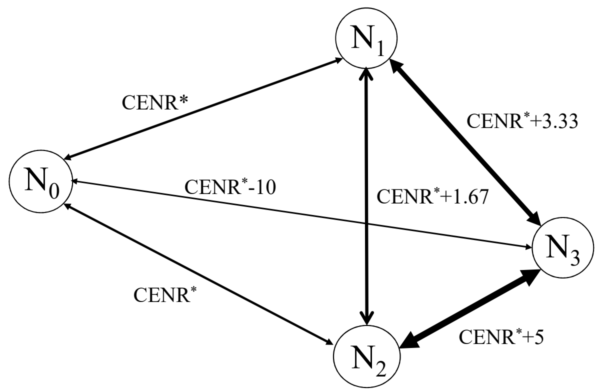


Figure 7.1: A broadcast network with one source (N_0) and three destinations (N_1, N_2, and N_3).

1. Whenever a node in the network obtains the file, it offers to become an intermediate source for its neighbors.

2. If a recipient has multiple neighbors willing to become intermediate sources, it chooses to receive from the neighbor from which it has the incoming link with the best *nominal* SNR.

Note that mode AS of network-coded broadcast described in Chapter 4 belongs to this class of broadcast file distributions. Therefore, an upper bound obtained for such broadcast can be used as a benchmark in the evaluation of CFC-AS or RLNC-AS. Furthermore, because our numerical results illustrate that mode AS of network-coded broadcast outperforms modes NS, TS, and CS, the bound is helpful in the evaluation of the latter three modes as well.

The session begins when node N_0 starts broadcasting packets to all three destinations. As the session progresses, one of five possible scenarios is encountered: (1) Node N_1

obtains the file first and starts sending packets to nodes N_2 and N_3. The first among N_2 and N_3 to obtain the file sends the remainder of the file to the other node. (2) N_2 obtains the file first and then sends packets to N_1 and N_3. If N_3 obtains the file before N_1 is able to do so, then N_3 starts sending packets to node N_1. Otherwise, N_2 keeps transmitting until both N_1 and N_3 get the file, (3) N_1 and N_2 obtain the file at the same time. Then N_2 starts sending packets to N_3. (4) N_3 obtains the file first and then sends packets to N_1 and N_2 until both nodes obtain the file. (5) All three destinations obtain the file at the same time.

The probability mass function for the session completion time N_S can be written as

$$P(N_S = n) = \sum_{i=1}^{5} P_i(n), \tag{7.2}$$

where $P_i(n)$ the probability that the ith scenario above is encountered and a total of n time units elapse from the beginning of the session until its completion. The expression for $P_1(n)$ is given by

$$\begin{aligned}
P_1(n) = \sum_{n_1=K}^{n} \sum_{k_1=0}^{K-1} \sum_{k_2=0}^{K-1} & P\left[T_{0,1}(K) = n_1, R_{0,2}(n_1) = k_1, R_{0,3}(n_1) = k_2\right] \\
& \times \Big\{ \sum_{n_2=K-k_1}^{n-n_1} \sum_{k_3=0}^{K-k_2-1} P[T_{1,2}(K-k_1) = n_2, R_{1,3}(n_2) = k_3, \\
& \qquad\qquad T_{2,3}(K-k_2-k_3) = n-n_1-n_2] \\
& \quad + \sum_{n_3=K-k_2}^{n-n_1} \sum_{k_4=0}^{K-k_1-1} P[T_{1,3}(K-k_2) = n_3, R_{1,2}(n_3) = k_4, \\
& \qquad\qquad T_{3,2}(K-k_1-k_4) = n-n_1-n_3] \\
& \quad + P\left[T_{1,2}(K-k_1) = n-n_1, T_{1,3}(K-k_2) = n-n_1\right] \Big\}.
\end{aligned} \tag{7.3}$$

The term $P\left[T_{0,1}(K) = n_1, R_{0,2}(n_1) = k_1, R_{0,3}(n_1) = k_2\right]$ immediately following the triple summation is the probability that it takes n_1 packet transmissions by N_0 to deliver the file to N_1 while N_2 and N_3 recover k_1 and k_2 of those transmissions, respectively. The three terms inside the curly braces are for the three situations that can occur after N_1 becomes the relay node and starts sending packets to N_2 and N_3. The first double summation is for the situation in which N_2 receives the file after N_1 makes n_2 transmissions, N_3 recovers k_3 packets from N_1 in the meantime, and receives the remaining packets from N_2. The second double summation is for the event that N_3 obtains the file from N_1 before N_2 is able to do so and then sends packets to N_2. The third term inside the braces is for the situation in which N_1 delivers the file to N_2 and N_3 at precisely the same time.

Because the links in the network are statistically independent, each joint probability in (7.3) can be written as a product of individual probabilities. Therefore, we rewrite (7.3) as follows:

$$
\begin{aligned}
P_1(n) = \sum_{n_1=K}^{n}\sum_{k_1=0}^{K-1}\sum_{k_2=0}^{K-1} & P[T_{0,1}(K)=n_1]\,P[R_{0,2}(n_1)=k_1]\,P[R_{0,3}(n_1)=k_2] \\
& \times\Big(\sum_{n_2=K-k_1}^{n-n_1}\ \sum_{k_3=0}^{K-k_2-1} \big\{ P[T_{1,2}(K-k_1)=n_2]\,P[R_{1,3}(n_2)=k_3] \\
& \qquad\qquad \times P[T_{2,3}(K-k_2-k_3)=n-n_1-n_2] \big\} \\
& \quad + \sum_{n_3=K-k_2}^{n-n_1}\ \sum_{k_4=0}^{K-k_1-1} \big\{ P[T_{1,3}(K-k_2)=n_3]\,P[R_{1,2}(n_3)=k_4] \\
& \qquad\qquad \times P[T_{3,2}(K-k_1-k_4)=n-n_1-n_3] \big\} \\
& \quad + P[T_{1,2}(K-k_1)=n-n_1]\,P[T_{1,3}(K-k_2)=n-n_1]\Big).
\end{aligned} \tag{7.4}
$$

Proceeding in a similar manner, we obtain the following expressions for $P_2(n)$

through $P_5(n)$:

$$P_2(n) = \sum_{n_1=K}^{n} \sum_{k_1=0}^{K-1} \sum_{k_2=0}^{K-1} P\left[T_{0,2}(K) = n_1\right] P\left[R_{0,1}(n_1) = k_1\right] P\left[R_{0,3}(n_1) = k_2\right]$$
$$\times \Big\{ \sum_{n_2=K-k_2}^{n-n_1} \sum_{k_3=0}^{K-k_1-1} P\left[T_{2,3}(K-k_2) = n_2\right] P\left[R_{2,1}(n_2) = k_3\right]$$
$$\times P\left[T_{3,1}(K-k_1-k_3) = n-n_1-n_2\right]$$
$$+ P\left[T_{2,1}(K-k_1) \leq n-n_1\right] P\left[T_{2,3}(K-k_2) = n-n_1\right] \Big\}. \tag{7.5}$$

$$P_3(n) = \sum_{n_1=K}^{n} \sum_{k_1=0}^{K-1} P\left[T_{0,1}(K) = n_1\right] P\left[T_{0,2}(K) = n_1\right] P\left[R_{0,3}(n_1) = k_1\right]$$
$$\times P\left[T_{2,3}(K-k_1) = n-n_1\right]. \tag{7.6}$$

$$P_4(n) = \sum_{n_1=K}^{n} \sum_{k_1=0}^{K-1} \sum_{k_2=0}^{K-1} P\left[T_{0,3}(K) = n_1\right] P\left[R_{0,1}(n_1) = k_1\right] P\left[R_{0,2}(n_1) = k_2\right]$$
$$\times P\left[\max\{T_{3,1}(K-k_1), T_{3,2}(K-k_2)\} = n-n_1\right]. \tag{7.7}$$

$$P_5(n) = P\left[T_{0,1}(K) = n\right] P\left[T_{0,2}(K) = n\right] P\left[T_{0,3}(K) = n\right]. \tag{7.8}$$

7.1.2 Multicast File Distribution via Relay Nodes

Consider the network of Figure 7.2 and suppose that node N_0 wants to transfer a file of K information packets to nodes N_1 and N_2. There is no usable direct link between N_0 and

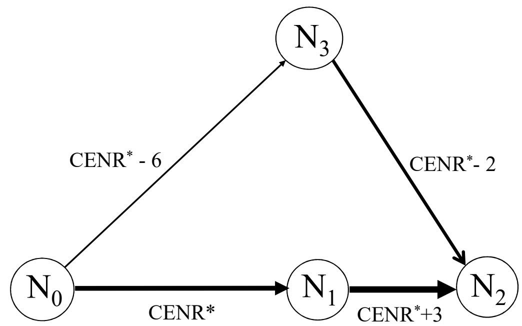


Figure 7.2: A multicast network in which source node N_0 wants to transfer a file to nodes N_1 and N_2.

N_2, but N_1 can obtain the file from N_0 and then forward the file to N_2. Node N_3 does not need the file but is willing to forward packets from N_0 to N_2. The session begins when N_0 starts sending packets to N_1. Node N_3 also tries to receive the packets transmitted by N_0. After recovering W packets, N_3 forwards the packets to N_2 and returns to the receive mode until it obtains another W packets. The process continues until N_1 obtains the file, instructs N_0 to cease transmission, and begins sending packets to N_2. The session concludes when N_2 obtains all packets in the file.

The strategy described above represents a class of multicast file-distribution techniques that includes mode PTA of network-coded multicast given in Chapter 6. Therefore, an upper bound on the session throughput for this class of multicast techniques can be used as a performance benchmark in the evaluation of CFC-PTA or RLNC-PTA.

The probability mass function for the session completion time N_S is given by

$$
\begin{aligned}
P(N_S = n) &= \sum_{\tilde{n}=K}^{n} \sum_{\tilde{k}=0}^{K} P\left[T_{0,1}(K) = \tilde{n}, Z_{3,2}(\tilde{n}) = \tilde{k}, T_{1,2}(K-\tilde{k}) = n-\tilde{n}\right] \\
&= \sum_{\tilde{n}=K}^{n} \sum_{\tilde{k}=0}^{K} P\left[T_{0,1}(K) = \tilde{n}\right] P\left[Z_{3,2}(\tilde{n}) = \tilde{k}\right] P\left[T_{1,2}(K-\tilde{k}) = n-\tilde{n}\right] \qquad (7.9)
\end{aligned}
$$

for $n \geq K$, where $Z_{3,2}(\tilde{n})$ is the number of packets delivered to N_2 by N_3 over a span of $\tilde{n}$ time units. The summand is the probability that node N_0 takes $\tilde{n}$ time units to deliver the file to N_1, node N_3 delivers $\tilde{k}$ packets to N_2 during that time, and then N_1 takes $n-\tilde{n}$ times units to deliver the remaining $K-\tilde{k}$ packets from the file to N_2. The summation is over all possible choices for $\tilde{n}$ and $\tilde{k}$. We have exploited the statistical independence of the links in going from the first line to the second in (7.9). The probability mass function for $Z_{3,2}(\tilde{n})$ is given by

$$
\begin{aligned}
P\left[Z_{3,2}(\tilde{n}) = \tilde{k}\right] &= \sum_{(\mathbf{n},\mathbf{k})} \prod_{i=1}^{p} P\left[T_{0,3}(W) = n_i\right] P[R_{3,2}(\min\{W, \tilde{n}-(i-1)W - \sum_{j=1}^{i} n_j\}) = k_i] \\
&+ \sum_{(\hat{\mathbf{n}},\hat{\mathbf{k}})} \{\prod_{i=1}^{q-1} P\left[T_{0,3}(W) = \hat{n}_i\right] P\left[R_{3,2}(W) = \hat{k}_i\right]\} P\left[T_{0,3}(W) > \hat{n}_q\right], \qquad (7.10)
\end{aligned}
$$

where the summations are over all possible vectors $(\mathbf{n},\mathbf{k}) = (n_1, k_1, n_2, k_2, \ldots, n_p, k_p)$ and $(\hat{\mathbf{n}},\hat{\mathbf{k}}) = (\hat{n}_1, \hat{k}_1, \hat{n}_2, \hat{k}_2, \ldots, \hat{n}_q, \hat{k}_q)$ for $p, q > 0$ that satisfy the following constraints:

$$
\begin{aligned}
&n_i, \hat{n}_i \geq W, \;\; \tilde{n} - pW \leq \sum_{i=0}^{p} n_i \leq \tilde{n} - (p-1)W, \;\; \sum_{i=0}^{p} k_i = \sum_{i=0}^{q} \hat{k}_i = \tilde{k}, \;\; \hat{k}_q = 0, \\
&\sum_{i=0}^{q} \hat{n}_i = \tilde{n} - (q-1)W, \;\; 0 \leq k_i \leq \min\{W, \tilde{n} - (i-1)W - \sum_{j=1}^{i} n_j\}.
\end{aligned}
$$

Each value of i in (7.10) represents a *round* that involves the recovery of W packets by N_3 from N_0 in n_i time units and the subsequent forwarding of those W packets by N_3 to N_2, of which k_i are recovered by the latter. The first summation is for those events in which the last round ends (i.e., N_1 instructs N_3 to stop forwarding) while N_3 is in the process of sending packets to N_2. The term $\min\{W, \tilde{n}-(i-1)W-\sum_{j=1}^{i} n_j\}$ inside the first summation accounts for the fact that, in the last round, there may not be enough time for N_3 to forward all W packets before N_1 starts transmitting. The second summation is for those events in which the last round ends while N_3 is in the process of receiving packets from N_0. When q is 1, the product term inside the curly braces must be interpreted as 1.

Valid choices for $(\mathbf{n},\mathbf{k})$ and $(\hat{\mathbf{n}},\hat{\mathbf{k}})$ include vectors of different lengths, i.e., the values of p and q are not necessarily the same for all $(\mathbf{n},\mathbf{k})$ and $(\hat{\mathbf{n}},\hat{\mathbf{k}})$ that satisfy the constraints above. However, p and q cannot exceed $\tilde{n}/2W$, which is the maximum number of rounds possible over a span of $\tilde{n}$ time units.

7.1.3 Two-Hop Relay Network

In the two-hop relay network of Figure 7.3, node N_0 wants to transfer a file of K information packets to node N_3 but the direct link between the two nodes is too poor to support packet delivery. Therefore, nodes N_1 and N_2, which do not need the file themselves, serve as relay nodes. N_0 begins the session by sending packets to N_1. When N_1 recovers W packets, it forwards those packets to N_3 by making W transmissions. When N_1 is transmitting to N_3, N_2 receives packets from node N_0. After forwarding W packets, N_1 returns to the receive mode while N_2 begins forwarding to N_3 the packets it recovered from N_0. The relays continue to alternate between receiving and transmitting until either N_3 or N_1 obtains the complete file. If N_3 obtains the file first, then the session ends. If N_1 obtains the file before N_3, then N_1 instructs N_0 and N_2 to stop transmitting, begins sending packets to N_3, and stops when N_3 receives the file.

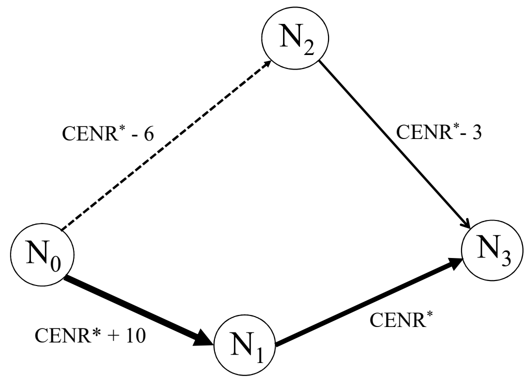


Figure 7.3: A two-hop relay network in which node N_0 wants to transfer a file to N_3 but does not have a direct link to it.

The strategy described above represents a class of file-distribution techniques that includes mode-DR of network-coded fie transfer in a two-hop relay network with one destination node. Hence, an upper bound for this class of methods can be used as a performance benchmark for CFC-DR or RLNC-DR. Because our numerical results showed that mode DR of network-coded file delivery outperforms mode LM, the results can also be used in the bounding of CFC-LM or RLNC-LM.

To simplify the analysis, we assume that K is an integer multiple of W. We describe the file transfer in terms of *phases*. In phase 1, node N_0 delivers W packets to node N_1 by making m_1 transmissions. Node N_2 stays idle during phase 1. During phase 2, N_1 sends W packets to N_3, of which k_1 are correctly delivered. In the meantime, N_2 recovers j_1 out of the W packets transmitted by N_0. In phase 3, N_1 recovers W packets from N_0 after the latter makes m_2 transmissions. In the meantime, N_2 transmits j_1 packets to N_3, of which l_1 are correctly decoded. Proceeding in this manner, we observe that the shortest possible session concludes at the end of phase $K/W+1$. We also notice that if the session concludes before or exactly at the end of phase $2K/W-1$, then N_1 does not get an opportunity to receive the complete file and become the sole transmitter for the remainder of the session. Otherwise, the session enters phase $2K/W$, in which N_1 instructs N_0 and N_2 to cease transmission and then sends packets to N_3 until N_3 obtains the complete file.

The probability mass function for the session completion time can be written as

$$P(N_S = n) = P_o(n) + P_e(n) + P_d(n). \tag{7.11}$$

The term $P_o(n)$ in (7.11) is the probability that the session ends during an odd-numbered phase (i.e., a phase in which N_2 transmits to N_3) and n time units elapse from the beginning of the session until its completion. $P_o(n)$ can be written as

$$\begin{aligned} P_o(n) = \sum_{(\mathbf{j},\mathbf{k},\mathbf{l},\mathbf{m})} \prod_{i=1}^{p} \{P[T_{0,1}(W) = m_i]\, P[R_{2,3}(j_{i-1}) = l_i]\, P[R_{1,3}(W) = k_i]\, P[R_{0,2}(W) = j_i]\} \\ \times P[R_{2,3}(n - pW - \sum_{r=1}^{p} m_r) = K - \sum_{r=1}^{p}(l_r + k_r)], \end{aligned} \tag{7.12}$$

where the summation is over all $(\mathbf{j},\mathbf{k},\mathbf{l},\mathbf{m}) = (j_1,\ldots,j_p,k_1\ldots,k_p,l_1,\ldots,l_p,m_1,\ldots,m_p)$ that satisfy the following constraints:
$j_i \le W$, $k_i \le W$, $l_i \le j_{i-1}$, $m_i \ge W$, and $0 < K - \sum_{r=1}^{p}(l_r + k_r) \le n - pW - \sum_{r=1}^{p} m_r \le j_{p-1}$.

Not all valid choices for the vector $(\mathbf{j},\mathbf{k},\mathbf{l},\mathbf{m})$ have the same length; however, they must satisfy $K/2W \le p < K/W$. In evaluating (7.12), we use $j_0 = 0$ and interpret $P(R_{2,3}(0) = l_i)$ to be 1 for $l_i = 0$, and 0 otherwise.

The term $P_e(n)$ in (7.11) is the probability that the session ends during an even-numbered phase (i.e., a phase in which N_1 transmits to N_3) but before phase $2K/W$ and n time units elapse from the beginning of the session until its completion. We can express

$P_e(n)$ as

$$P_e(n) = \sum_{(\mathbf{j},\mathbf{k},\mathbf{l},\mathbf{m})} \prod_{i=1}^{p-1} \{P[T_{0,1}(W) = m_i] P[R_{2,3}(j_{i-1}) = l_i]$$
$$\times P[R_{1,3}(W) = k_i] P[R_{0,2}(W) = j_i]\}$$
$$\times P\left[R_{2,3}(j_{p-1}) = l_p\right] P[T_{0,1}(W) = m_p]$$
$$\times P[R_{1,3}(n-(p-1)W - \sum_{r=1}^{p} m_r) = K - \sum_{r=1}^{p-1}(l_r + k_r) - l_p], \quad (7.13)$$

where the first four constraints on $(\mathbf{j},\mathbf{k},\mathbf{l},\mathbf{m})$ are the same as those for (7.12) and the fifth constraint is $0 < K - \sum_{r=1}^{p-1}(l_r + k_r) - l_p \le n - (p-1)W - \sum_{r=1}^{p} m_r \le W$.

The term $P_d(n)$ in (7.11) is the probability that N_1 obtains the file before N_3 obtains it, the session enters phase $2K/W$, and n time units elapse since the beginning of the session until its completion. The expression for $P_d(n)$ is

$$P_d(n) = \sum_{(\mathbf{j},\mathbf{k},\mathbf{l},\mathbf{m})} \prod_{i=1}^{K/W-1} \{P[T_{0,1}(W) = m_i] P[R_{2,3}(j_{i-1}) = l_i]$$
$$\times P[R_{1,3}(W) = k_i] P[R_{0,2}(W) = j_i]\}$$
$$\times P\left[R_{2,3}(j_{K/W-1}) = l_{K/W}\right] P\left[T_{0,1}(W) = m_{K/W}\right]$$
$$\times P[T_{1,3}(K - \sum_{r=1}^{K/W-1} (l_r + k_r) - l_{K/W}) = n - K + W - \sum_{r=1}^{K/W} m_r]. \quad (7.14)$$

Again, the first four constraints are the same as those for (7.12) and (7.13). The fifth constraint is $0 < K - \sum_{r=1}^{K/W-1}(l_r + k_r) - l_{K/W} \le n - K + W - \sum_{r=1}^{K/W} m_r$.

7.2 Evaluation of the Bounds

To compute the bounds, we first evaluate the appropriate expression for $P\left(N_S = n\right)$ from Section 7.1.3, obtain the average session completion time $\overline{N}_S$, and then substitute the value of $\overline{N}_S$ in (7.1). The probability mass functions for $R_{i,j}(n)$ and $T_{i,j}(k)$, which are required in the evaluation of $P\left(N_S = n\right)$, depend on the statistical characteristic of the link between the nodes. For static AWGN links, packet erasures are independent Bernoulli events. Consequently, $R_{i,j}(n)$ and $T_{i,j}(k)$ have the well-known binomial and negative binomial distributions, respectively, and it is straightforward to compute $P\left(N_S = n\right)$.

If, on the other hand, the links have correlated block fading or shadowing in addition to thermal noise, packet erasures on a link are correlated and the evaluation of $P\left(N_S = n\right)$ is difficult in general. But the situation is relatively easy to handle for a network in which the time-varying propagation loss can be modeled by a two-state Markov chain and a capacity-achieving channel code is used for each packet transmission. The probability mass function for $T_{i,j}(k)$ for the link between node i and node j in such a network is given by [26]

$$\begin{aligned} P\left[T_{i,j}(k) = n\right] = {} & \tilde{p}_{i,j}^{-1}(l_{i,j} - p_{i,j})\tilde{q}_{i,j}^{n-2k+1} \\ & \times \sum_{s=0}^{k-1} \binom{k-1}{s} \binom{s+n-k-1}{n-k} (\tilde{p}_{ij} q_{i,j})^s (p_{i,j} - q_{i,j})^{k-s-1} \\ & + \tilde{p}_{ij}^{-1} \tilde{l}_{ij} \tilde{q}_{ij}^{n-2k} \sum_{s=0}^{k} \binom{k}{s} \binom{s+n-k-1}{n-k} (\tilde{p}_{ij} q_{i,j})^s (p_{i,j} - q_{i,j})^{k-s} \end{aligned} \tag{7.15}$$

for $n \geq k$, where $l_{i,j}$ is the probability that the first packet transmitted on the link is successfully decoded, $p_{i,j}$ is the probability that decoding of a packet succeeds given that the previous packet was successfully decoded, and $q_{i,j}$ is the probability that decoding of a packet succeeds given that the previous packet was erased. Also, $\tilde{l}_{i,j} = 1 - l_{i,j}$, $\tilde{p}_{i,j} = 1 - p_{i,j}$, and $\tilde{q}_{i,j} = 1 - q_{i,j}$. The probability mass function for $R_{i,j}(n)$ is given by

$$P\left[R_{i,j}(n) = k\right] = P\left[T_{i,j}(k) \leq n\right] - P\left[T_{i,j}(k+1) \leq n\right]. \tag{7.16}$$

The probabilities $l_{i,j}$, $p_{i,j}$, and $q_{i,j}$ can be determined from the capacity limit Γ for the code-modulation combination used, the nominal CENR of the link from node i to node j, the transition probabilities of the two-state Markov chain that models the link, and the channel gains associated with the states of the Markov chain. Let $\text{CENR}^*_{i,j}$ be the nominal CENR of the link from node i to node j and let the channel gains associated with the two states, numbered 0 and 1, be η_0 and η_1, respectively, where $\eta_0 < \eta_1$. If $\text{CENR}^*_{i,j} + \eta_0 < \Gamma$ and $\text{CENR}^*_{i,j} + \eta_1 \geq \Gamma$, then a packet transmitted by node i is recovered by node j if the link between the nodes is in state 1, and is erased otherwise. Assuming that the Markov chain is in steady state, $l_{i,j} = \pi(1)$, $p_{i,j} = p'(1|1)$, and $q_{i,j} = p'(1|0)$, where $\pi(s)$ denotes the steady-state probability of state s and $p'(\hat{s}|s)$ is the probability of a transition from state s to state $\hat{s}$ in one step. If $\text{CENR}^*_{i,j} + \eta_s \geq \Gamma$ for both $s = 0$ and $s = 1$, then packet erasures never occur on the link. For such links, $P\left[T_{i,j}(k) = n\right] = P\left[R_{i,j}(n) = k\right] = 1$ for $n = k$, and 0 otherwise. If $\text{CENR}^*_{i,j} + \eta_s < \Gamma$ for both $s = 0$ and $s = 1$, then all packets transmitted on the link are erased. For such links, $P\left[R_{i,j}(n) = k\right] = 1$ for $k = 0$, and 0 otherwise. The probability $P\left[T_{i,j}(k) = n\right]$ is undefined in this case and can be handled by letting $P\left[T_{i,j}(k) = n\right] = 0$ for all finite, non-zero values of k and n in the analysis.

For all results in this section, we consider a file consisting of 100 information packets of 3249 bits each and assume that a channel code of rate 0.793 is used along with QPSK modulation. The capacity limit for this combination is 4 dB. The two-state Markov chain that models the propagation loss has parameters $p'(0|1) = 0.002$, $p'(1|0) = 0.01$,

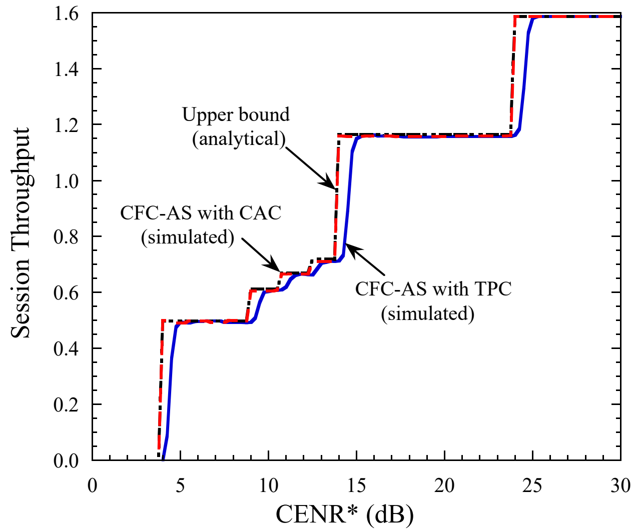


Figure 7.4: Comparison of the session throughput of CFC-AS in the network of Figure 7.1 with the upper bound.

$\pi(0) \approx 0.167$, $\pi(1) \approx 0.833$, $\eta_0 = -10$ dB, and $\eta_1 = 0$ dB.

Figure 7.4 shows the upper bound obtained for broadcast file distribution in the network of Figure 7.1 using the method of Section 7.1.1. The bound is compared with the simulated throughput of file transfer in the same network when mode AS of CFC-based broadcast described in Chapter 4 is employed. A TPC of rate 0.793 along with QPSK is used for transmissions in CFC-AS. We observe that fountain-coded broadcast lags the upper bound by less than a dB. Two factors contribute to this difference: the excess packets required for fountain decoding and the suboptimality of the TPC relative to the hypothetical CAC that is used in the evaluation of the bound. Figure 7.4 also shows the simulated throughput of CFC-AS when a CAC of rate 0.793 along with QPSK modulation is used for transmissions. It can be seen that the throughput of CFC-AS when the CAC is used for transmissions is virtually identical to the upper bound. This implies that the gap between the upper bound and the throughput of CFC-AS with the TPC is almost entirely due to the difference between the performance of the CAC and the TPC.

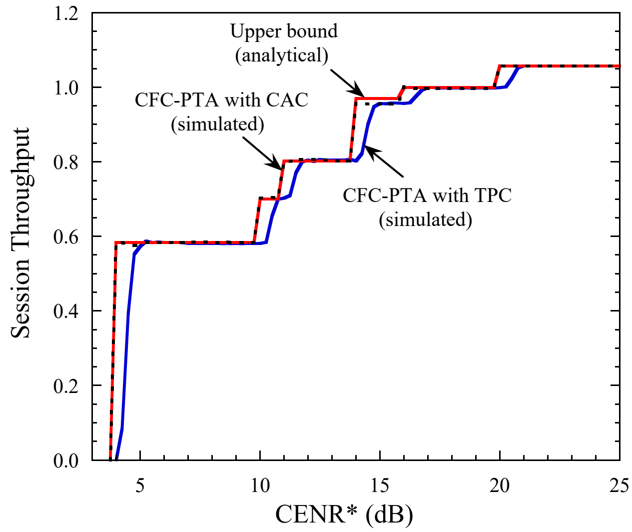


Figure 7.5: Comparison of the session throughput of CFC-PTA in the network of Figure 7.2 with the upper bound.

In Figure 7.5, we plot the upper bound obtained for multicast file distribution in the network of Figure 7.2 using the method given in Section 7.1.2. The parameter W is set to 25. The bound is compared with the throughput of CFC-PTA from Chapter 6 with $W_a = W$, node N_1 as the primary relay, and node N_3 as the secondary relay. Note that, for this network, operation of mode PTB is also the same as that of mode PTA. Because N_1 is a destination, it acts as a TAD relay and sends packets only after it has obtained the file. Only N_3 is available to forward batches of packets from node N_0 to N_2 before receiving the complete file. We observe from Figure 7.5 that the throughput of CFC-PTA when a CAC of rate 0.793 is used along with QPSK for transmissions is very close to the upper bound. The throughput of CFC-PTA when the CAC is replaced by a TPC of the same rate is within 0.7 dB of the upper bound.

Figure 7.6 shows the upper bound obtained for the two-hop relay network of Figure 7.3 using the method of Section 7.1.3 with $W = 25$. The bound is compared with the throughput of the CFC-DR strategy from Chapter 5 that employs the same value of W and

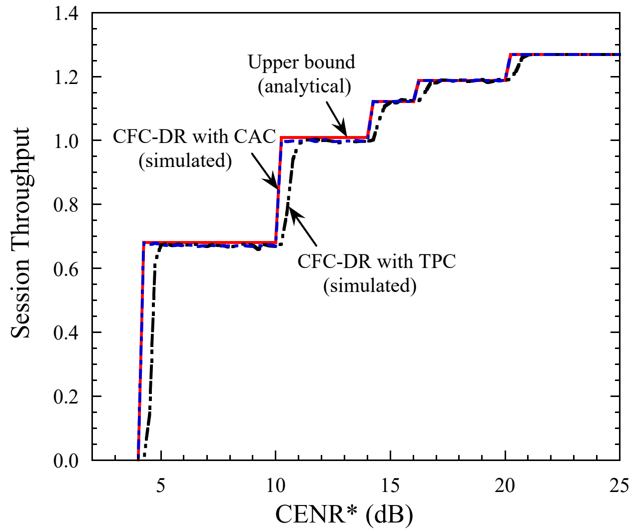


Figure 7.6: Comparison of the session throughput of CFC-DR in the network of Figure 7.3 with the upper bound.

designates N_1 as the primary relay and N_2 as the secondary relay. Note that the throughput of CFC-DR is also the throughput of CFC-PTB because the two methods operate identically for this network. As before, the throughput of CFC-DR when a CAC of rate 0.793 is used along with QPSK for transmissions is almost indistinguishable from the upper bound. The throughput of CFC-DR with a TPC of the same rate is within 0.7 dB of the bound.

Chapter 8

Conclusion

Network coding can greatly improve the throughput of file distribution in ad hoc packet radio networks. We described and evaluated strategies for network-coded broadcast and multicast distribution of files in networks of half-duplex nodes. In our methods, nodes exploit the broadcast nature of the wireless medium by opportunistically listening to transmissions made by senders other than their designated relays, multiple relay nodes cooperate to deliver files to one or more destinations, and duplicate packets are either completely avoided or rarely produced.

Our techniques can be implemented using either fountain coding or random linear network coding. For fountain-coded file delivery, we introduced the concept of continued fountain coding, a mechanism which ensures that duplicate fountain-coded packets are never generated by a relay node that applies fountain coding to a decoded file. A relay node that applies random linear network coding to a file, on the other hand, rarely produces duplicate packets even when no special measures are employed. With either form of network coding, our strategies significantly outperform conventional file-transfer schemes that use ARQ for retransmissions.

We showed that fountain coding provides slightly higher throughput than random linear network coding. Two factors contribute to this difference in throughput. First, random linear network coding incurs some overhead because it requires each transmitted packet to include an encoding vector, which conveys the identities of the constituent information packets to the recipient. Secondly, random linear network coding divides the file into generations to reduce the computational complexity at the cost of an increased number of excess packets. For fountain coding, it suffices to include only a sequence number in each transmitted packet instead of appending an encoding vector. Also, the availability of lower-complexity decoding techniques for fountain coding means that a file of moderate size need not be divided into generations.

We derived analytical upper bounds on the session throughput of file transfers in networks consisting of four half-duplex nodes. We evaluated the bounds for networks in which the links experience time-varying propagation loss modeled by a two-state Markov chain. Our suggested methods for network-coded file distribution were found to perform very close to the bounds.

Appendices

Appendix A List of Abbreviations

The following is a list of abbreviations used in the dissertation, grouped according to the context in which they appear:

- ☐ Network coding
  - ○ FC: Fountain coding
  - ○ CFC: Continued fountain coding
  - ○ RLNC: Random linear network coding
  - ○ NC packet: Network-coded packet
- ☐ Modes of operation and relay functionality
  - ○ Broadcast networks
    - NS: No secondary
    - TS: Temporary secondary
    - CS: Choose secondary
    - AS: All secondary
  - ○ Two-hop relay networks
    - LM: Low memory
    - DR: Network decoding at relays
    - FR: Forwarding by relays
    - NR: Network coding at relays
  - ○ General multicast networks
    - TAD: Transmit after decoding the file

- TBD: Transmit before decoding the file
- PTA: Primary relays operate in the TAD mode
- PTB: Primary relays operate in the TBD mode
- PTB-PCK: Mode PTB with perfect channel knowledge

☐ Generation selection for RLNC

- RS: Random selection
- ID: Increment after decoding
- RR: Round robin

☐ Modulation formats

- BOK: Biorthogonal key
- BPSK: Binary phase-shift key
- QPSK: Quadriphase shift key
- QAM: Quadrature amplitude modulation

☐ Other abbreviations

- AMCC: Adaptive modulation and channel coding
- ARQ: Automatic repeat request
- AWGN: Additive white Gaussian noise
- CAC: Capacity-achieving code
- CENR: Chip-energy-to-noise-density ratio
- DRR: Data-recovery rate
- ETX: Expected transmission count

- Max-DRR: Maximum data-recovery rate
- Min-index: Minimum suggested index
- SDRR: Single-transmission data-recovery rate
- SNR: Signal-to-noise ratio
- TPC: Turbo product code

Appendix B Probabilistic Model for the Raptor Decoder

We employ a probabilistic model derived from the results in [4] to simulate the decoding of the raptor code. In [4], it is given that if raptor coding is applied to a file of K information packets and if j raptor-coded packets are available at the input of the decoder, then the probability that raptor decoding fails (i.e., the probability that the number of innovative packets is less than K) can be approximated by

$$\beta[j,K] \approx \begin{cases} 1, & j < K, \\ 0.85(0.567)^{j-K}, & j \geq K. \end{cases} \tag{B.1}$$

In our implementation, the recipient tries to decode the raptor code as soon as it recovers the Kth packet and continues to do so for each newly recovered packet until decoding succeeds. Therefore, we are interested in the probability that fountain decoding fails after recovering the ith fountain-coded packet ($i \geq K$) given that it has failed for each of the previously recovered packets. Let this probability be denoted by $P_f[i,K]$. It follows from (B.1) that $P_f[i,K] \approx 0.85$ for $i = K$. For $i > K$, a straightforward application of Bayes' rule gives

$$\begin{aligned} P_f[i,K] &\approx \frac{\beta[i,K]}{\beta[i-1,K]} \\ &= \frac{0.85(0.567)^{i-K}}{0.85(0.567)^{i-K-1}} \\ &= 0.567. \end{aligned} \tag{B.2}$$

Therefore, the probability that fountain decoding succeeds upon the recovery of the ith fountain-coded packet ($i \geq K$) after having failed for each of the previously received packets can be expressed as

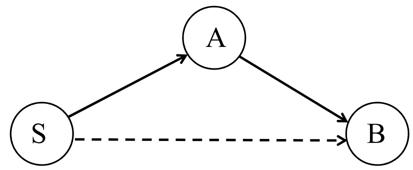


Figure B.1: A three-node network.

$$
\begin{aligned}
P_s[i,K] &= 1 - P_f[i,K] \\
&\approx \begin{cases} 0.15, & i = K, \\ 0.433, & i > K. \end{cases}
\end{aligned} \tag{B.3}
$$

Because we employ a systematic raptor code, decoding succeeds with probability 1 if the recipient is able to recover each of the fountain-coded packets with sequence numbers 0 through $K-1$. Otherwise, after the recipient recovers the ith NC packet ($i \geq K$), our simulation draws a Bernoulli random variable X such that $P(X=1) = P_s[i,K]$. The raptor-decoding attempt is declared a success if $X = 1$. Otherwise, a decoding failure occurs and the recipient waits to recover another NC packet before trying to decode the file again.

To verify the accuracy of the approximation, we simulate a CFC-based broadcast session in the network of Figure B.1. Node S has a file consisting of 500 information packets that must be transferred to nodes A and B. Each link in the network experiences Rayleigh fading. The nominal CENR of the links S–A and A–B is CENR* whereas that nominal CENR of the link S–B is CENR$^* - 10\,$dB. Node A acts as a relay between S and B; however, B tries to overhear the channel packets that S sends to A. When A is able to decode the file, it instructs S to cease transmissions, applies continued fountain coding to the information packets, and starts transmitting fountain-coded packets to B. In

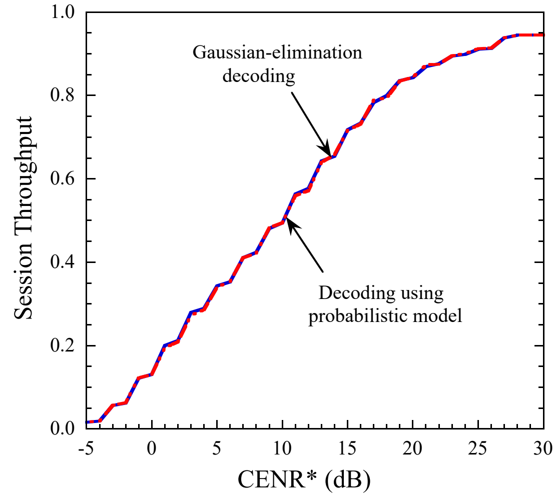


Figure B.2: Comparison of Gaussian-elimination decoding and the probabilistic decoding model for CFC-based broadcast.

Figure B.2, we compare the numerical results obtained by using the probabilistic model for raptor decoding with the results obtained when the raptor code is decoded using Gaussian elimination. All transmissions use a turbo product code of rate 0.472 as the channel code and the channel packets are modulated using QPSK. We observe that the two curves are almost indistinguishable, which demonstrates the accuracy of the probabilistic model.

Appendix C Adaptive Modulation and Channel Coding

To combat the effects of time-varying propagation loss on the links, a sender may adapt the modulation format and the channel-coding rate from one channel packet to the next by means of an adaptive modulation and channel coding (AMCC) protocol. Let $\mathcal{B} = \{B(j) : 1 \leq j \leq N\}$ be the set of code-modulation combinations available for use and suppose that the elements of $\mathcal{B}$ are indexed in order of increasing *information rate*, where the information rate $\rho(j)$ for combination $B(j)$ is defined as the number of information bits per modulation chip when that combination is used. When the radio receiver at a recipient demodulates and decodes a channel packet, it calculates a statistic referred to as the *error count*. The error count is the number of binary symbol errors obtained when hard decisions are made at the output of the demodulator. Many iterative decoder modules provide the value of the error count along with the decoded information bits. If the decoder does not provide this information, the error count can be computed by encoding the decoded word with the same channel code that was used for transmission and then comparing the resulting codeword bit-by-bit with the hard-decision demodulator output. The number of bits that do not match is the error count. Next, an interval test is applied to the error count to obtain a *suggested index*, which is the index of the code-modulation combination the recipient would like for the sender to use for the next transmission. If the received packet used combination $B(j)$ and the error count is in the interval $I_j(l)$, then $B(l)$ is the recipient's suggested index for the next packet. The intervals are given by $I_j(l) = [\gamma_{j,l}, \gamma_{j,l-1})$ for $j = 1, \ldots, N$ and $1 \leq l \leq N$.

The recipient conveys the suggested index to the sender by including it in a feedback message. For unicast transmissions in our network-coded file distributions, a recipient may either report its suggested index after every packet or it may report the index after every vth packet ($v > 1$) to restrict the number of feedback messages from growing very large.

The sender stores the most recent suggested index provided by the recipient and uses the code-modulation combination with that index for transmitting channel packets until a new suggested index arrives. As shown in [27], only a small reduction in throughput is incurred if the recipient provides intermittent feedback with $v = 10$ instead of sending feedback for every recovered packet.

The error count, and consequently the suggested index, can be obtained only if a packet is decoded correctly. If a packet fails to decode, resulting in a lack of acknowledgement or a negative acknowledgment from the recipient, the sender lowers the recipient's suggested index by one for the next packet. However, if the failed packet was transmitted using the lowest-rate combination (i.e., the recipient's most recent suggested index was 1), then no change is made.

For network-coded multicast transmissions, the AMCC protocol employs a round-robin reporting strategy [5] which ensures that at most one recipient sends a feedback message in response to a transmitted channel packet. In the header of each transmitted packet, the sender specifies which recipient should send feedback for that packet. Recipients are chosen in a round-robin fashion. The sender maintains a table that stores the most recent suggested index from each recipient.

The *maximum data-recovery rate* (max-DRR) criterion [5] is used to select a combination for the next packet based on the entries in the table. The objective of the max-DRR criterion is to maximize the *single-transmission data recovery rate* (SDRR) for the next packet. The SDRR is the ratio of the number of information bits recovered from a packet by all recipients combined to the number of time units required to transmit that packet. Therefore, if D destinations are able to decode a packet that uses combination B_j, then the SDRR for that packet is $\rho_j D$.

Let $\tilde{I}_r$ be the most recent suggested index received from destination r. The Max-DRR protocol expects that the destination will be able to decode any channel packet that uses a code-modulation combination whose index does not exceed $\tilde{I}_r$. Thus, the number of destinations that can decode a channel packet with combination $B(j)$ is $W(j) = |\{r : \tilde{I}_r < j\}|$. Hence, the expected SDRR for combination $B(j)$ is $R(j) = \rho(j)W(j)$. The max-DRR protocol chooses combination $B(m)$ for the next packet if

$$R(m) = \max\{R(j) : 1 \leq j \leq N\}. \tag{C.1}$$

In more than one combination achieves the maximum, than the combination with the highest information rate $\rho(j)$ is chosen.

When ARQ is employed instead of network coding, the recipient must acknowledge each received packet. Therefore, we assume that a recipient of ARQ-based transmissions reports its suggested index to the sender after every received packet by including the index in the acknowledgement message. Even for ARQ-based multicast transmissions, each recipient acknowledges every received packet and sends the suggested index along with the acknowledgement. Adaptation is performed using the *min-index* criterion [5], in which the index for the code-modulation combination for the next packet is the smallest of the suggested indexes. A combination with a smaller index has a lower rate and a higher probability of packet recovery; therefore, min-index reduces the probability that the packet will have to be retransmitted.

The AMCC protocol in our simulations employs the set of 13 code-modulation combinations listed in Table 1. The modulation formats are 64-biorthogonal key (64-BOK), binary phase-shift key (BPSK), quadriphase shift key (QPSK), and 16-quadrature amplitude modulation (16-QAM). The set of channel codes for the AMCC protocol consists of five turbo product codes [23] of rates 0.260, 0.346, 0.472, 0.620, and 0.766. The interval endpoints used by the AMCC protocol to choose a suggested index for the next packet are listed in Table 2. Notice that only a subset of the possible 20 code-modulation combinations

Table 1: Code-Modulation Combinations

j	Modulation	Code rate
1	64-BOK	0.260
2	64-BOK	0.472
3	64-BOK	0.766
4	BPSK	0.260
5	BPSK	0.346
6	QPSK	0.260
7	QPSK	0.346
8	QPSK	0.472
9	QPSK	0.620
10	QPSK	0.766
11	16-QAM	0.472
12	16-QAM	0.620
13	16-QAM	0.766

are used for transmissions. Our approach to choosing an appropriate subset is described in [25]. The selection criterion is based on the *single-packet throughput* of a combination on a static AWGN channel. The single-packet throughput for code-modulation combination $B(j)$ is given by $S_p(j)=\rho(j)P_c(j)$, where $P_c(j)$ is the probability that the packet using combination $B(j)$ is decoded correctly. For a given set of code-modulation combinations, some combinations may not provide higher single-packet throughput than any of the other combinations in the set at any SNR of interest. Such combinations can be eliminated without affecting the performance of the AMCC protocol.

Table 2: Interval Endpoints.

l	$\gamma_{1,l}$	$\gamma_{2,l}$	$\gamma_{3,l}$	
1	1221	673		
2	425	235	145	
3	88	49	30	
4	0	0	0	
l	$\gamma_{4,l}$	$\gamma_{5,l}$	$\gamma_{6,l}$	$\gamma_{7,l}$
3	1904			
4	1588	1194		
5	1079	811	1844	
6	781	1588	1526	1148
7	519	390	429	907
8	211	159	727	547
9	81	61	429	323
10	0	0	159	120
11			28	21
12			0	0
l	$\gamma_{8,l}$	$\gamma_{9,l}$	$\gamma_{10,l}$	
7	665			
8	401	305		
9	237	180	147	
10	88	67	55	
11	0	0	0	
l	$\gamma_{11,l}$	$\gamma_{12,l}$	$\gamma_{13,l}$	
10	663			
11	422	321		
12	236	179	146	
13	0	0	0	

Appendix D A Method for Selecting Relay Nodes

A strategy for selecting relay nodes for network-coded file distribution is outlined in this appendix. Our approach, which is based on the concept of least-resistance routing [28], involves the construction of a source-specific minimum-cost multicast tree to designate the primary relays followed by the identification of the next higher-cost paths to assign the secondary relays.

Let the nodes in a network be indexed $\mathrm{N}_0, \mathrm{N}_1, \ldots, \mathrm{N}_{M-1}$, where M is the number of nodes. Consider a route or a path in the network that consists of H hops or links. We consider link *costs* or *resistances* of the form $r_k = \alpha + \beta e_k$, where α, β are non-negative constants and e_k represents some measure of the quality of the kth link. The parameter e_k is such that a smaller value indicates a better link. The resistance of the path is given by

$$R = \sum_{k=1}^{H} r_k = \alpha H + \beta \sum_{k=1}^{H} e_k. \tag{D.1}$$

The constants α and β can be tuned to adjust the relative emphasis on the number of hops and the quality of the links. Note that setting $\beta = 0$ results in a hop-count based algorithm that does not consider link resistances.

Depending on the application, the value of e_k may represent different link characteristics, including signal-to-noise ratio, probability of packet erasures, probability of bit error, and expected transmission count. In our illustrations in Chapters 4 and 6, e_k is the number of test-symbol errors observed on the kth link over a period of time. We assume that the nodes in the network periodically exchange control packets and that each control packet includes 128 binary test symbols known a priori at all nodes. The first 64 test symbols are modulated using QPSK and the remaining 64 symbols are modulated using 16-QAM. Each node demodulates the test symbols it receives from its neighbors and records the number of binary symbol errors. At the time of tree construction, the fraction of symbols that were in error over the last 25 test-symbol sequences received over the kth link is used as the value of e_k.

The path metric given by (D.1) is used to construct a spanning tree such that each destination is connected to the source via the least-resistance path. Many practical algorithms are available to construct such trees [24]. The intermediate nodes in the spanning tree thus obtained are the primary relays in our applications.

The next task is to assign the secondary relays. To find a secondary relay for node N_i, we first determine the set Ω_i, which consists of all nodes that are not primary relays and are fewer hops away from the source than node N_i is. If Ω_i is an empty set, then no secondary relay is assigned to node N_i. Otherwise, node $N_{j'}$ is the secondary relay for node N_i if

$$j' = \arg\min_{j \in \Omega_i} \; R_j + r_{j,i}, \tag{D.2}$$

where R_j is the resistance of the path from the source to node N_j on the spanning tree and $r_{j,i}$ is the resistance of the link from node N_j to node N_i. For mode PTB of network-coded multicast, additional constraints are imposed on the selection of secondary relays as explained in Chapter 6. While assigning secondary relays for this mode, the set Ω_i must be chosen such that each node in the set satisfies the constraints.

Appendix E Generation Selection for Random Linear Network Coding

As mentioned in Chapter 2, a file may be divided into g disjoint generations of d packets each to reduce the computational complexity at the RLNC decoder. For each NC packet, the sender must first choose a generation and then combine packets from that generation. In this appendix, we examine three approaches to choosing the generation. In the following, we assume that the generations are indexed 1 through g.

- *Random selection* (RS): For each NC packet, an integer G is drawn at random according to a uniform distribution on the set of integers $\{1, 2, \ldots, g\}$. The NC packet is formed by applying RLNC to the information packets in the generation with index G.

- *Increment after decoding* (ID): The sender starts a session by applying RLNC to packets in generation 1 and proceeds to the next generation only after the current generation has been decoded by all recipients.

- *Round-robin selection* (RR): A parameter $l > 0$ is employed by this method. The sender chooses generations in a round-robin fashion, sending l consecutive NC packets from a generation before proceeding to the next. Once K transmissions have been made, l is set to 1.

For a numerical evaluation of these approaches, we consider a multicast session in which a source sends a file of 500 information packets to 15 destinations using RLNC. The file is divided into 5 generations of 100 packets each. The source sends packets directly to the destinations; i.e., no relaying is involved. The link from the source to each destination has a nominal CENR of CENR* and experiences Rayleigh fading with a normalized

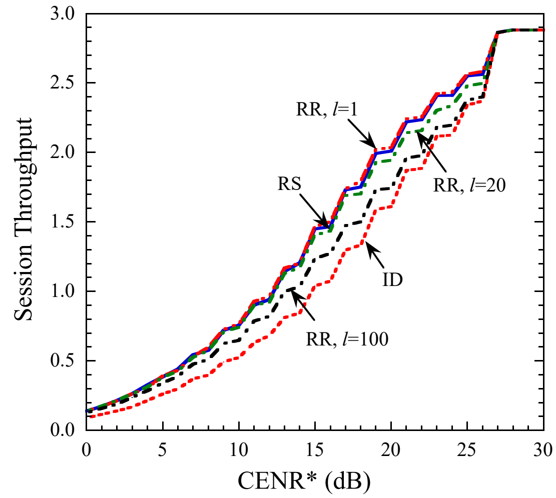


Figure E.1: Comparison of different methods for generation selection for RLNC-based multicast transmissions from a source to 15 destinations over Rayleigh-fading links.

Doppler frequency of 0.02. The throughput of the file transfer is shown in Figure E.1. We notice that the RR method with $l = 1$ provides the best performance, followed very closely by the RS strategy. The throughput of RR drops as l is made larger. The performance of ID is the worst among the three. We also observe that all strategies give the same performance when the SNR is high enough such that there are no packet erasures on the links. It is only at lower SNR that there is a difference between the throughput of different strategies.

The difference in the throughput of the strategies is due to the fact that, for some strategies, packet erasures at different destinations are confined to different generations. In such situations, many of the NC packets transmitted by the source are from generations that one or more destinations have already decoded. This leads to an increase in the decoding completion times at the destinations and lowers the throughput. In contrast, if the erasures at each destination are spread out over all generations, almost all transmissions made by the source is of interest to every destination and the decoding completion times are lower. Recall that in our model for fading, the fade levels on a link are correlated whereas the fading processes on different links are independent. A consequence of correlated fading is that packet erasures at each receiver tend to occur in bursts. When large number of consecutive NC packets from the same generations are transmitted by the sender (as in ID, or in RR with a large value of l), a recipient may experience bursts of erasures that are confined to a generation. Due to the statistical independence of the fading from link to link, different recipients may experience error bursts in different generations. In contrast, the RS strategy and the RR strategy with a small value of l have an interleaving-type effect, which causes the erasures at the recipients to be distributed over all generations and improves the session throughput.

References

[1] M. Luby, L. Vicisano, J. Gemmell, L. Rizzo, M. Handley, and J. Crowcroft, "The use of forward error correction (FEC) in reliable multicast," IETF Request for Comments 3453, December 2002.

[2] D. J. C. MacKay, "Fountain codes," *IEE Proceedings–Communications*, vol. 152, no. 6, pp 1062–1068, December 2005.

[3] T. Ho and D. S. Lun, *Network coding: An introduction*, Cambridge University Press, Cambridge, U.K., 2008.

[4] M. Luby, T. Gasiba, T. Stockhammer, and M. Watson, "Reliable multimedia download delivery in cellular broadcast networks," *IEEE Transactions on Broadcasting*, vol. 53, no. 1, pp. 235–246, March 2007.

[5] J. D. Ellis and M. B. Pursley, "Adaptive transmission protocols for fountain-coded multicast transmissions in packet radio networks," *IEEE Transactions on Communications*, vol. 65, no. 4, pp. 1786–1796, April 2017.

[6] A. F. Molisch, N. B. Mehta, J. S. Yedidia, and J. Zhang, "Performance of fountain codes in collaborative relay networks," *IEEE Transactions on Wireless Communications*, vol. 6, no. 11, pp. 4106–4119, November 2007.

[7] A. Nessa, M. Kadoch, R. Q. Hu, and B. Rong, "Towards reliable cooperative communications in clustered ad hoc networks," *Proceedings of the IEEE Global Communications Conference*, pp. 4090–4095, December 2012.

[8] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," *Proceedings of the ACM SIGCOMM*, pp. 169–180, August 2007.

[9] D. Koutsonikolas, Y.-C. Hu, and C.-C. Wang, "Pacifier: High-throughput, reliable multicast without "crying babies" in wireless mesh networks," *IEEE/ACM Transactions on Networking*, vol. 20, no. 5, pp. 1375–1388, October 2012.

[10] S. S. Borkotoky and M. B. Pursley, "Preliminary results on the performance of an adaptive protocol for packet relay with fountain coding," *Proceedings of the IEEE Military Communications Conference*, pp. 362–367, October 2014.

[11] S. S. Borkotoky, M. C. Dowling, and M. B. Pursley, "Some results on two forms of erasure-correction coding for packet radio networks," *Proceedings of the IEEE Information Theory and Applications Workshop*, pp. 265–269, February 2015.

[12] S. S. Borkotoky and M. B. Pursley, "Broadcast file distribution in a four-node packet radio network with network coding and code-modulation adaptation," *Proceedings of the IEEE Military Communications Conference*, pp. 1144–1149, October 2015.

[13] S. S. Borkotoky and M. B. Pursley, "Network-coded file distribution in an ad hoc relay network," *Proceedings of the IEEE Information Theory and Applications Workshop*, DOI 10.1109/ITA.2016.7888194, February 2016.

[14] S. S. Borkotoky and M. B. Pursley, "A method for network-coded broadcast in an ad hoc Network," *Proceedings of the IEEE Military Communications Conference*, pp. 418–423, November 2016.

[15] S. S. Borkotoky and M. B. Pursley, "A comparison of two methods for fountain-coded file distribution in an ad hoc network with relays," *Proceedings of the IEEE Information Theory and Applications Workshop*, February 2017.

[16] M. Luby, "LT codes," *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science*, pp. 271–280, 2002.

[17] A. Shokrollahi, "Raptor codes," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2551–2567, June 2006.

[18] M. Luby, A. Shokrollahi, M. Watson, and T. Stockhammer, "Reliable forward error correction scheme for object delivery," IETF Request for Comments 5053, October 2007.

[19] M. Nakagami, "The m-distribution – A general formula of intensity distribution of rapid fading," in *Statistical Methods in Radio Wave Propagation*, W. C. Hoffman (ed.), pp. 3–36, Pergamon Press, London, 1960.

[20] M. A. Juang and M. B. Pursley, "Finite-state Markov chain models for the intensity of Nakagami fading," *International Journal of Wireless Information Networks*, vol. 20, no. 2, pp. 95–102, May 2013.

[21] G. L. Stuber, *Principles of Mobile Communication* (2nd Ed.), Kluwer, Norwell, MA, 2001.

[22] M. K. Simon and M.-S. Alouini, *Digital Communications over Fading Channels* (2nd Ed.), Wiley, Hoboken, NJ, 2005.

[23] Advanced Hardware Architectures, Inc., Product Specification for AHA4501 Astro 36 Mbits/sec Turbo Product Code Encoder/Decoder. Available: http://www.aha.com

[24] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, "Energy-Efficient Broadcast and Multicast Trees in Wireless Networks," *Mobile Networks and Applications*, vol. 7, no. 6, pp. 481–492, 2002.

[25] S. S. Borkotoky and M. B. Pursley, "Applications of capacity limits to performance analyses of adaptive transmission protocols for packet radios," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 1, pp. 71–78, January 2017.

[26] R. Viveros, K. Balasubramanian and N. Balakrishnan, "Binomial and negative binomial analogues under correlated Bernoulli trials," *The American Statistician*, pp. 243–247, vol. 48, no. 3, August 1994.

[27] J. D. Ellis and M. B. Pursley, "Integration of adaptive modulation and channel coding with fountain coding for packet radio systems," *IEEE Transactions on Communications*, vol. 63, no. 5, pp. 1510–1521, May 2015.

[28] M. B. Pursley and H. B. Russell, "Network protocols for frequency-hop packet radios with decoder side information," *IEEE Journal on Selected Areas in Communications*, vol. 12, no. 4, pp. 612–621, May 1994.